

Exploring the Potential of Multimodal and Multiphasic Brain- Computer Interfaces for Neurorehabilitation



Trinity College Dublin

Coláiste na Tríonóide, Baile Átha Cliath

The University of Dublin

Colin Simon, M.Sc.

A thesis submitted for the degree of

Doctor in Philosophy

in the School of Psychology, Trinity College Dublin, Ireland July 2023

Supervisor

Dr Kathy Ruddy

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

This work includes the unpublished and published work of others, duly acknowledged in the text wherever included.

I agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

I consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

Signed,

A handwritten signature in dark ink, appearing to be 'Colin Simon', written in a cursive style.

Colin Simon

27th July 2023

Summary

Stroke is the second biggest contributor to disability adjusted life years in Europe (Feigin et al., 2021). Ongoing demographic shifts (United Nations, 2017) will increase stroke incidence (Lawrence et al., 2001; Nakayama et al., 1994; Virani et al., 2020) and subsequently add to adult disability (Virani et al., 2020). The simultaneous decline in stroke related fatalities results in an increase of survivors grappling with disability (Vos et al., 2020) amplifying the need for effective and user friendly poststroke motor rehabilitation therapies.

Since there is likely a period of heightened brain plasticity shortly after stroke, therapies should begin as early as possible (Krakauer & Carmichael, 2017; Stinear et al., 2020). However, options for motor rehabilitation during this time period are limited as many therapies require a minimum amount of functional movement (Kwakkel et al., 2015; Winstein et al., 2016). Motor Imagery (MI) based Brain-Computer Interfaces (BCIs) using Electroencephalography (EEG) scalp recorded signals, enable early poststroke motor rehabilitation by encouraging the maintenance of neural activity in damaged motor pathways.

In this thesis we investigated why, despite encouraging results, EEG BCIs are not widely used for motor rehabilitation post-stroke. The two main issues we identified are the lack of scalability and the lack of justifiability. Scalability issues are impacted by the cost of treatment that renders BCI inefficient in medical practice. Justifiability issues are rooted in the insufficient mechanistic understanding of the control mechanisms of BCI and how their use could contribute to recovery. Therefore, we suggest to focus on the working mechanisms of BCI to increase medical justifiability. Further, we suggest splitting up BCI systems to specifically exploit mechanisms that make BCI work (BCI working mechanisms) or address specific rehabilitative challenges as well as investigating the contributions of variables specifically linked to BCI use and other nonspecific variables.

To mitigate some of the identified prohibiting factors, we designed a BCI that has two distinct phases using two modalities (Transcranial Magnetic Stimulation Neurofeedback, TMS-NF, and EEG-BCI), that both operate based on feedback of the neural activity associated with MI. We chose specific MI instructions that optimally provoke neural mechanisms linked to poststroke motor recovery (Stinear et al., 2015) and that are quick for BCI users to learn (Liang et al., 2020; Ruddy et al., 2018). We investigated whether pre-training (priming) with TMS-NF would improve subsequent performance on an EEG BCI.

We found that participants pre-trained with TMS-NF were faster and more accurate at learning how to control the EEG BCI than the participants training with only an EEG BCI.

Further, we found that the TMS-NF group produced MI patterns resembling Motor Execution (ME), a property not seen in the EEG BCI only group. While these results increased justifiability by suggesting a possible medically viable mechanism, scalability was not improved as the TMS-NF and the EEG BCI require trained administrators.

To increase scalability we designed a reduced version of the EEG BCI that stroke survivors could use at home without supervision. We mailed EEG kits and instructions to the homes of healthy participants, who were able to set up, train with, and achieve control over the BCI system.

The next steps for the research programme involve testing TMS-NF with stroke patients in a Randomised Controlled Trial (RCT) to investigate potential benefits for upper limb function. Due to the COVID-19 Pandemic, the research programme was put on hold for two years and data from the next phase of the experiment will not be complete until late 2024.

In the intervening period, we explored how the unique attributes of BCI could potentially expand to other conditions. In Body Integrity Disphoria (BID), patients suffer from a mismatch between physical and mental body representation. In our work we summarised the current state of BID research and suggest several designs for BCI-based therapies.

As the stroke RCT will involve analysis of neuroimaging data in elderly patients to investigate neural recovery mechanisms, we made use of an existing large dataset, the Irish Longitudinal Study of Ageing (TILDA) to explore how white matter connectivity in elderly brains relates to motor behaviour- in this case motor inhibition and falling. Based on the result of this study, we hypothesised that there may be some value in designing a BCI specifically targeting motor inhibition in elderly people to reduce falls.

In this thesis, we addressed some of the shortcomings of existing BCI approaches outlined in Simon et al. (2021) by designing a multiphase BCI that is faster to learn, and further developed it by adding a scalable at home training phase (Simon & Ruddy, 2022). This work was shortlisted for the prestigious international BCI Award 2022. With this body of work we may have contributed to the ongoing shift in the discussion around medical BCI from basic research and technology development to the design of translational therapies (Colucci et al., 2022; Losanno et al., 2023; Remsik et al., 2022).

List of Publications and Presentations

Publications

Chapter 2

Simon, C., Bolton, D. A. E., Kennedy, N. C., Soekadar, S. R., & Ruddy, K. L. (2021). Challenges and Opportunities for the Future of Brain-Computer Interface in Neurorehabilitation. *Frontiers in Neuroscience*, 15, 814.
<https://doi.org/10.3389/fnins.2021.699428>

Chapter 4

Simon, C., & Ruddy, K. L. (2022). A wireless, wearable Brain-Computer Interface for neurorehabilitation at home; A feasibility study. 2022 10th International Winter Conference on Brain-Computer Interface (BCI), 1–6.
<https://doi.org/10.1109/BCI53720.2022.9734849>

Chapter 6

Chakraborty, S., Saetta, G., Simon, C., Lenggenhager, B., & Ruddy, K. (2021). Could Brain–Computer Interface Be a New Therapeutic Approach for Body Integrity Dysphoria? *Frontiers in Human Neuroscience*, 15.
<https://www.frontiersin.org/articles/10.3389/fnhum.2021.699830>

Prepared for Submission

Chapter 3

Chapter 5

Chapter 7

Presentations

2023 BCI Award Podcast – Neurocareers: Doing the Impossible!

2022 BCI Award nominee – Presentation

2022 Brainbox Initiative Conference – Poster

2022 Young Investigator Symposium – Poster

2022 8th Science Factory: TMS–EEG Summer School and Workshop – 10 minute presentation

2022 3rd BCI-UC – 20 Minute Presentation – Won a 20 minute presentation slot

2022 10th International Winter Conference on Brain-Computer Interface (BCI) – Poster

2021 Brainbox Initiative Conference – Poster – Won Poster Prize

2021 Neuroscience Ireland Young Investigator Symposium – Poster

2020 Brainbox Initiative Conference – Poster

2020 Neuroscience Ireland Young Investigator Symposium – Poster

Acknowledgements

I would like to thank everyone who has played a role in the completion of my doctoral thesis. It has been a long, unpredictable journey, and many people have been vital to my success.

First, I want to acknowledge the Defense Committee, Professor Doctor Paul Dockree, and Doctor Mahnaz Arvanez. Their guidance and expertise have helped me refine my research.

Doctor Kathy Ruddy, my supervisor, friend, and mentor, your input and advice have been invaluable. You have provided both professional guidance and personal support throughout the process.

Thanks also to the Progress Committee, including Professor Doctor Robert Whelan, Doctor Maeve Bracken, and Professor Doctor Ladislav Timulak, your feedback and encouragement have helped me to stay focused and motivated.

I'm grateful to the Health Research Board (HRB), Enterprise Ireland (EI), and Trinity College Dublin, for their funding and resources.

I would like to extend my thanks to my peers Jiv, Nadja and Daniel for all the lunch breaks, coffee breaks, virtual coffee breaks, gossip breaks, and of course, sometimes, regular breaks. Thanks for helping me take breaks and keeping me motivated to give my best.

During this journey, I've had the pleasure of working with some very talented people like Olga, Lamia, Helene, Rebecca and all our research students. Working alongside such smart and talented people has been a rewarding experience.

To all my study participants, thank you for your time and contribution. Your input has been essential to the outcome of this thesis.

I appreciate the friendship and support from Noah, Rike, Benji, Mark and David who have provided much-needed distraction and relief in between the intense periods of research.

A special acknowledgment to Vivienne, who somehow found time to marry me in between all the proofreading, editing, emotional support giving and all the other things she did with me along this journey. Similarly, I would like to thank my family for their consistent support and visits. Thank you for being there for me when it counts. Lastly, I want to thank my dog Juska who kept my feet warm during the long nights of writing.

Thank you all.

Table of Contents

DECLARATION.....	3
SUMMARY	4
LIST OF PUBLICATIONS AND PRESENTATIONS	6
PUBLICATIONS	6
<i>Chapter 2</i>	<i>6</i>
<i>Chapter 4</i>	<i>6</i>
<i>Chapter 6</i>	<i>6</i>
PREPARED FOR SUBMISSION.....	6
<i>Chapter 3</i>	<i>6</i>
<i>Chapter 5</i>	<i>6</i>
<i>Chapter 7</i>	<i>6</i>
PRESENTATIONS	6
ACKNOWLEDGEMENTS	8
TABLE OF CONTENTS.....	9
LIST OF FIGURES	14
LIST OF TABLES.....	17
LIST OF EQUATIONS	19
LIST OF ABBREVIATIONS.....	20
1. GENERAL INTRODUCTION.....	24
2. BRAIN-COMPUTER INTERFACES IN NEUROREHABILITATION .	30
2.1 CHALLENGES AND OPPORTUNITIES FOR THE FUTURE OF BRAIN-COMPUTER INTERFACE IN NEUROREHABILITATION.....	30
2.1.1 <i>Abstract.....</i>	<i>30</i>
2.1.2 <i>Introduction.....</i>	<i>30</i>
2.1.3 <i>Brain-Computer Interface for Neurorehabilitation; Basic Premise and Scope of the Review</i>	<i>31</i>
2.1.4 <i>Practical and Technical Challenges With Clinical Implementation of BCI.....</i>	<i>32</i>
2.1.5 <i>Challenges for Elucidating Mechanisms Underlying BCI-Induced Functional Improvements</i>	<i>35</i>

2.1.6	<i>Outlook for Future Scalability and Justification of BCI Use Clinically..</i>	37
2.1.7	<i>Conclusions</i>	38
2.1.8	<i>Author Contributions</i>	39
2.1.9	<i>Conflict of Interest</i>	39
2.1.10	<i>Acknowledgments</i>	39
3.	TRANSCRANIAL MAGNETIC STIMULATION NEUROFEEDBACK – A MULTIMODAL, MULTIPHASE APPROACH TO STROKE REHABILITATION USING BCI	40
3.1	CHAPTER INTRODUCTION.....	40
3.2	OPTIMISING CONTROL OF MOTOR IMAGERY-BASED BCIs BY PRIMING USING TMS-NEUROFEEDBACK.....	40
3.2.1	<i>Abstract</i>	40
3.2.2	<i>Introduction</i>	41
3.2.3	<i>Methods</i>	44
3.2.4	<i>Results</i>	58
3.2.5	<i>Discussion</i>	85
3.2.6	<i>Author Contributions</i>	93
4.	THE FEASIBILITY OF MULTIPHASE BCI REHABILITATION	94
4.1	CHAPTER INTRODUCTION.....	94
4.2	A WIRELESS, WEARABLE BRAIN-COMPUTER INTERFACE FOR NEUROREHABILITATION AT HOME; A FEASIBILITY STUDY	94
4.2.1	<i>Abstract</i>	94
4.2.2	<i>Introduction</i>	95
4.2.3	<i>Methods</i>	95
4.2.4	<i>Results</i>	101
4.2.5	<i>Discussion</i>	105
4.2.6	<i>Author Contributions</i>	110
5.	CONTEMPORARY CONSIDERATIONS FOR BCI DESIGN IN NEUROREHABILITATION	111
5.1	CHAPTER INTRODUCTION.....	111
5.2	KEEPING OUR EYES ON THE PRIZE; ARE WE LOSING SIGHT OF THE ‘WHY’ IN BCI FOR NEUROREHABILITATION?	111

5.2.1	<i>Identification of Appropriate Target Neural Signal</i>	114
5.2.2	<i>Identifying Therapeutic Goals</i>	115
5.2.3	<i>Identifying the Most Appropriate Technology to Improve Function..</i>	115
5.2.4	<i>Thinking Outside the (Black) Box</i>	116
5.2.5	<i>Conclusion</i>	117
5.2.6	<i>CRediT Contributions</i>	117
6.	APPLICATIONS OF BCI FOR RARE DISEASES AND DISORDERS	118
6.1	CHAPTER INTRODUCTION	118
6.2	COULD BRAIN–COMPUTER INTERFACE BE A NEW THERAPEUTIC APPROACH FOR BODY INTEGRITY DYSPHORIA?.....	118
6.2.1	<i>Abstract</i>	118
6.2.2	<i>Introduction</i>	118
6.2.3	<i>Characteristics and Aetiology of BID</i>	119
6.2.4	<i>Current Approaches to Treatment for BID</i>	124
6.2.5	<i>Structural and Functional Connectivity Abnormalities in BID</i>	126
6.2.6	<i>A New Generation of Research into BID Treatment Using Neurofeedback / Brain-Computer Interface</i>	127
6.2.7	<i>Conclusion</i>	130
6.2.8	<i>Author Contributions</i>	130
6.2.9	<i>Funding</i>	131
6.2.10	<i>Conflict of Interest</i>	131
6.2.11	<i>Publisher’s Note</i>	131
7.	STRUCTURAL BRAIN CONNECTIVITY IN ELDERLY INDIVIDUALS RELATING TO MOTOR INHIBITION NETWORKS	132
7.1	CHAPTER INTRODUCTION	132
7.2	STOPPING NETWORKS AND FALLING IN THE ELDERLY: RELATING WHITE MATTER ARCHITECTURE IN THE BRAIN’S INHIBITORY CONTROL NETWORKS TO REAL- WORLD METRICS OF FALLING IN ELDERLY ADULTS	132
7.2.1	<i>Abstract</i>	132
7.2.2	<i>Introduction</i>	133
7.2.3	<i>Methods</i>	135
7.2.4	<i>Results</i>	141
7.2.5	<i>Discussion</i>	152
7.2.6	<i>Conclusion</i>	154

7.2.7	<i>Author Contribution</i>	155
8.	GENERAL DISCUSSION	156
8.1	CHALLENGES AND OPPORTUNITIES	156
8.2	MULTIPHASE AND MULTIMODAL BCI	156
8.3	THE FEASIBILITY OF MULTIPHASE AND MULTIMODAL BCI.....	157
8.4	DESIGNING BCI FOR REHABILITATION	158
8.5	BCI FOR RARE DISEASES AND DISORDERS.....	159
8.6	MOTOR INHIBITION AND FALLING IN ELDERLY PEOPLE	159
8.7	CONTEXT	160
8.8	STRENGTH AND LIMITATIONS	161
8.9	OUTLOOK.....	162
8.10	CONCLUSION.....	163
9.	REFERENCES	164
10.	APPENDICES	203
10.1	OPTIMISING BCI CONTROL	203
10.1.1	<i>MEP Model Assumptions</i>	203
10.2	STOPPING NETWORKS AND FALLING IN THE ELDERLY, APPENDIX.....	230
10.2.1	<i>Table 1. ROI Comparison</i>	230
10.2.2	<i>Assumptions for logistic regression</i>	230
10.2.3	<i>Results for control area</i>	234
10.2.4	<i>Results for apparent fibre density</i>	237
10.2.5	<i>Results for fractional anisotropy</i>	239
10.2.6	<i>Results for Apparent Fibre Density with Physical Activity interaction</i> ...	242
10.2.7	<i>Results for Fractional Anisotropy with Physical Activity interaction</i>	244
10.3	ETHICS APPROVAL	246
10.4	CODE	247
10.4.1	<i>Useful Functions (R)</i>	247
10.4.2	<i>MEP Check Script (MATLAB)</i>	262
10.4.3	<i>EEG Signal Analysis (MATLAB)</i>	266
10.4.4	<i>Automatic Preprocessing of EEG Data using ASR and CSD (MATLAB)</i>	273
10.4.5	<i>Stability and Distinctiveness (MATLAB)</i>	279
10.4.6	<i>QS Data (R)</i>	283

<i>10.4.7 TILDA Plot (R)</i>	285
<i>10.4.8 Classification Accuracy (R)</i>	300
<i>10.4.9 Power 1 (R)</i>	307
<i>10.4.10 Power 2 (R)</i>	309
<i>10.4.11 Power 3 (R)</i>	314
<i>10.4.12 CSP Processing (R)</i>	323
<i>10.4.13 MEP Hypothesis Testing (R)</i>	335
<i>10.4.14 MEP Preprocessing (R)</i>	344
<i>10.4.15 MEP Plots (R)</i>	355
<i>10.4.16 Stability and Distance (R)</i>	361
<i>10.4.17 Stability and Distance 2 (R)</i>	364

List of Figures

Figure 3.1. Timeline of the experiment.....	46
Figure 3.2. A short visual summary of TMS-NF.....	48
Figure 3.3. A short visual summary of the EEG BCI.....	51
Figure 3.4. EEG Epochs.....	54
Figure 3.5. MEP Amplitude over Time	59
Figure 3.6. Comparison first EEG session.....	65
Figure 3.7. Comparison session 3	66
Figure 3.8. Comparison session 5	67
Figure 3.9. Bimodal Distribution of Distinctiveness Up vs Down	69
Figure 3.10. CSP weight distributions.....	71
Figure 3.12. Relative Power High Alpha Split.....	75
Figure 3.13. Relative Power Comparison TMS vs EEG Split	76
Figure 3.14. Percentage Change High Alpha Split Comparison Group and Control	78
Figure 3.15. Percentage Change High Alpha Split Comparison Control.....	79
Figure 3.16. Percentage Change High Beta Split Comparison Control and Group	81
Figure 3.17. Percentage Change Low Beta Split Comparison Control Group	83
Figure 3.18. Percentage Change Low Beta Split Control	84
Figure 4.1. EEG BCI Protocol	98
Figure 4.2. Classification Accuracy per Participant.....	103
Figure 4.3. Mean Alpha Power Between Electrodes, Conditions and MI	105
Figure 5.1. TMS-NF Setup.....	112
Figure 7.1. Research Gap	135
Figure 7.2. Regions of Interest and Reconstructed Streamlines	139
Figure 7.3 AFD Scores Split by BP	144
Figure 7.4. AFD Values by Group and Age.....	145

Figure 7.5. rIFG to rSTN AFD Values by Group and Age.....	147
Figure 7.6. rIFG (Subregion) to rSTN AFD Values by Group and Age.....	149
Figure 10.1. Assumptions Basic MEP Model.	203
Figure 10.2. Assumptions MEP Model with Session Variable.	204
Figure 10.3. Assumptions MEP Model with BrMS.	205
Figure 10.4. Assumptions BrMS Model.	206
Figure 10.5. Assumptions Basic CA Model.	207
Figure 10.6. Assumptions MEP Model Using KVIQ and Session variables.	208
Figure 10.7. Assumptions Basic CA Model.....	209
Figure 10.8. Assumptions CA Model Using KSS.....	210
Figure 10.9. Assumptions Using KSS Full Interaction.....	211
Figure 10.10. Assumptions Rest Model Using KSS.....	212
Figure 10.11. Assumptions Distinctiveness Model.....	213
Figure 10.12. Relative Power Low Alpha Dotplot.....	214
Figure 10.13. Relative Power Low Alpha Boxplot.....	214
Figure 10.14. Relative Power Low Gamma Dotplot	215
Figure 10.15. Relative Power Low Gamma Boxplot.....	216
Figure 10.16. Relative Power High Delta Dotplot.....	217
Figure 10.17. Relative Power High Delta Boxplot	217
Figure 10.18. Relative Power Low Beta Dotplot.....	218
Figure 10.19. Relative Power Low Beta Boxplot	219
Figure 10.20. Percentage Change High Delta Dotplot.....	220
Figure 10.21. Percentage Change High Delta Boxplot.....	220
Figure 10.22. Percentage Change Low Alpha Dotplot	221
Figure 10.23. Percentage Change Low Alpha Boxplot.....	222
Figure 10.24. Percentage Change Theta Dotplot	223
Figure 10.25. Percentage Change Theta Boxplot.....	223

Figure 10.26. Percentage Change Low Beta Dotplot.....	224
Figure 10.27. Percentage Change Low Beta Boxplot	225
Figure 10.28. Percentage Change High Beta Dotplot	226
Figure 10.29. Percentage Change High Beta Boxplot	226
Figure 10.30. Percentage Change Low Gamma Dotplot	227
Figure 10.31. Percentage Change Low Gamma Boxplot.....	228
Figure 10.32. Relative Power Theta Dotplot.....	229
Figure 10.33. Relative Power Theta Boxplot.....	229
Figure 10.34. rIFG to STN Normal DistributionAFD Values.....	231
Figure 10.35. rIFG Subregion to STN Normal DistributionAFD Values	231
Figure 10.36 Linear relation rIFG	232
Figure 10.37. Linear Relation rIFG Subregion	233

List of Tables

Table 3.1 <i>Variables of the MEP Amplitude Model</i>	60
Table 3.2 <i>Variables of the MEP Amplitude Model, adjusted for different sessions</i>	60
Table 3.3 <i>Variables of the MEP Amplitude Model, adjusted for Background Muscle Activity (BrMS)</i>	61
Table 3.4 <i>Variables of the model predicting Background Muscle Activity (BrMS)</i> .	62
Table 3.5 <i>Variables of the MEP Amplitude Model, adjusted for Background Muscle Activity (BrMS)</i>	62
Table 3.6 <i>Classification Accuracy Model</i>	63
Table 3.7 <i>Classification Accuracy Model, adjusted by KSS Scale</i>	63
Table 3.8 <i>Classification Accuracy Model, Adjusted by KSS Scale and KSS interactions</i>	64
Table 3.9 <i>Model Predicting the Stability of Rest</i>	68
Table 3.10 <i>Model Predicting CA Using the Distinctiveness Between the MI State of Up and Down</i>	69
Table 3.11 <i>Powerband Split</i>	72
Table 3.12 <i>Summarised Results of Powerbands</i>	73
Table 3.13 <i>Model Predicting CA Using High Alpha (11-14Hz) Relative Power</i>	74
Table 3.14 <i>Model Predicting CA Using Low Alpha (11-14Hz) Percentage Change</i>	77
Table 3.15 <i>Model Predicting CA Using High Beta (22-31Hz) Percentage Change</i>	80
Table 3.16 <i>Model Predicting CA Using High Beta (22-31Hz) Percentage Change</i>	82
Table 6.1 <i>Taxonomy of Disorders Related to Body Integrity Dysphoria</i>	122
Table 7.1 <i>Basic Demographic Variables of Participants at Wave 3</i>	142
Table 7.2 <i>Association Between Microstructural Integrity in Inhibitory Control Networks and Odds of Falls in the Elderly</i>	143
Table 7.3 <i>Association Between Microstructural Integrity in Inhibitory Control Networks and Odds of Falls in Older Adults</i>	146

Table 7.4 <i>Association Between Microstructural Integrity in Inhibitory Control Networks and Odds of Falls in Older Adults</i>	148
Table 7.5 <i>Prediction of Fall Risk in Older Adults by White Matter Microstructure</i>	150
Table 7.6 <i>Association Between Microstructural Integrity in Inhibitory Control Networks and Odds of Falls in Older Adults</i>	151
Table 7.8 <i>Association Between Microstructural Integrity in Inhibitory Control Networks and Odds of Falls in Older Adults</i>	151
Table 7.9 <i>Association Between Microstructural Integrity in Inhibitory Control Networks and Odds of Falls in Older Adults</i>	152
Table 10.1 <i>CA Model Low Alpha</i>	213
Table 10.2 <i>CA Model Low Gamma</i>	215
Table 10.3 <i>CA Model High Delta</i>	216
Table 10.4 <i>CA Model Low Beta</i>	218
Table 10.5 <i>CA Model High Delta</i>	219
Table 10.6 <i>CA Model Low Alpha</i>	221
Table 10.7 <i>CA Model Theta</i>	222
Table 10.8 <i>CA Model Low Beta</i>	224
Table 10.9 <i>CA Model High Beta</i>	225
Table 10.10 <i>CA Model Low Gamma</i>	227
Table 10.11 <i>CA Model Theta</i>	228
Table 10.12 <i>ROI Voxel Count Comparison</i>	230
Table 10.13 <i>ROI Cases</i>	233
Table 10.14 <i>Results for Control Area FA and AFD</i>	234
Table 10.15 <i>Results for ROI, AFD</i>	237
Table 10.16 <i>Results for ROI, FA</i>	239
Table 10.17 <i>Results for AFD with Physical Activity</i>	242
Table 10.18 <i>Results for FA with Physical Activity</i>	244

List of Equations

Equation 1. Classification Accuracy Registered Model	52
Equation 2. Classification Accuracy Base Model	53
Equation 3. MEP Amplitude Base Model	53
Equation 4. EOG Correction	54
Equation 5. Percentage Change	55
Equation 6. Relative Bandpower	56
Equation 7. Stability	56
Equation 8. Distinctiveness	57
Equation 9. BrMS Base Model.....	61
Equation 10. Classification Accuracy EEG Power Model	72

List of Abbreviations

ADHD	Attention Deficit Hyperactivity Disorder
ADM	Abductor Digiti Minimi
AFD	Apparent fibre density
AHS	Anarchic Hand Syndrome
AR	Augmented Reality
ASR	Artifact Subspace Reconstruction
BCI	Brain-Computer Interface
BID	Body Integrity Dysphoria
BIID	Body-Integrity Identity Disorder
BP	Blood Pressure
CA	Classification Accuracy
CAMI	Centre for Advanced Medical Imaging
CI	Confidence Interval
CIMT	Constraint Induced Movement Therapy
CNS	Central Nervous System
CSP	Common Spatial Patterns
Down	Motor imagery instruction corresponding to a state of Downregulation of corticospinal excitability
DWI	Diffusion Weighted Imaging
EEG	Electroencephalography
EMG	Electromyogram

EOG	Electrooculoencephalogram
FA	Fractional Anisotropy
FDI	First Dorsal Interosseus
FFA	Fusiform Face Area
fMRI	Functional Magnetic Resonance Imaging
fNIRS	Functional Near Infrared spectroscopy
FOD	Fibre Orientation Distribution
GAD	Generalised Anxiety Disorder
GD	Gender Incongruity/Dysphoria
GDPR	General Data Protection Regulations
Hz	Hertz
ICA	Independent Component Analysis
IFC	Inferior Frontal Cortex
IFG	Inferior Frontal Gyrus
IPAQ	International Physical Activity Questionnaire
IPL	Inferior Parietal Lobule
IQR	Interquartile Range
KSS	Karolinska Sleepiness Scale
KVIQ	Kinesthetic and Visual Imagery Questionnaire
LDA	Linear Discriminant Analysis
l(ROI)	Region of Interest in the left brain hemisphere
MD	Mean Diffusivity

ME	Motor Execution
MEG	Magnetoencephalogram
MEP	Motor Evoked Potential
MI	Motor Imagery
mm	Millimeter
MMSE	Mini Mental State Examination
MoCA	Montreal Cognitive Assessment
MRI	Magnetic Resonance Imaging
MT-BCI	Mental Task Brain-Computer Interface
PCA	Principal Component Analysis
preSMA	presupplementary Motor Area
PTSD	Posttraumatic Stress Disorder
RCT	Randomised Controlled Trial
RD	Radial Diffusivity
Rest	State of rest with no motor imagery instruction
rMS or BrMS	(Background) Root Mean Square (of Muscle Activity)
rMT	Resting Motor Threshold
ROI	(Brain) Region of Interest
r(ROI)	Region of Interest in the right brain hemisphere
s	Second
S1	Primary Sensorimotor Area

SJH/AMNCH	St James's Hospital/Adelaide and Meath Hospital, Inc. National Children's Hospital
SPL	Superior Parietal Lobule
SPP	Somatoparaphrenia
SSRI	Selective Serotonin Reuptake Inhibitors
STN	Subthalamic Nucleus
TMS	Transcranial Magnetic Stimulation
TMS-NF	Transcranial Magnetic Stimulation Neurofeedback
Up	Motor imagery instruction corresponding to a state of upregulation of corticospinal excitability

1. General Introduction

In Europe 9 million people were living with stroke in 2017, and 1.5 million people were newly diagnosed with stroke (Luengo-Fernandez et al., 2020). These numbers will increase (Lawrence et al., 2001; Nakayama et al., 1994; Virani et al., 2020) due to ongoing demographic shifts (United Nations, 2017) and subsequently add to adult disability (Virani et al., 2020). Furthermore, the undoubtedly positive decline in stroke related fatalities has in turn increased the number of survivors grappling with disabilities (Vos et al., 2020) leading to an increased need for poststroke motor rehabilitation therapies.

However, advances in motor rehabilitation after stroke have been slow. The observation that the majority of stroke survivors regain around 70% of their potential recovery within three to six months of the incident is called the proportional recovery rule (Prabhakaran et al., 2008; Winters et al., 2015). This rule has been criticised (Hawe et al., 2019; Hope et al., 2019), but the strong correlation between the impairment immediately after the stroke and the impairment three to six months later as well as continued observation of this proportional recovery rule given the heterogeneity of stroke rehabilitation, implies that “The rehabilitation received between these two time points is having no impact on recovery from impairment” (Krakauer & Carmichael, 2017, p. 69). According to Krakauer and Carmichael (2017), the conclusion is that the proportional recovery rule is capturing a spontaneous biological recovery process that is not affected by existing rehabilitation approaches. Motor recovery after stroke progresses in a nonlinear, logarithmic fashion (Langhorne et al., 2011) and that it is likely that humans benefit from a phase of increased plasticity after stroke (Krakauer & Carmichael, 2017; Stinear et al., 2020), however, most poststroke motor recovery trials are performed after this brief period of enhanced biological plasticity. Motor recovery after this period is possible, but ignoring the brief window of biological recovery is a missed opportunity to gain some much-needed efficiency and a better understanding of positive plasticity and how to support it. These controversies can be generalised to research on motor rehabilitation after stroke.

Currently, Constraint Induced Movement Therapy (CIMT) is a well-researched therapy approach that comes closest to being a gold standard (Winstein et al., 2016). However, researchers are divided on the additional benefits of CIMT over dose-matched, task-based training tailored to the individual’s needs (Abdullahi et al., 2021; Corbetta et al., 2015; de Azevedo et al., 2022; Kwakkel et al., 2015; Rocha et al., 2021). Additionally, CIMT requires stroke survivors to be capable of producing a sufficient level of functional movement; This is a significant drawback as this level of movement is rarely present in the

early phases of stroke recovery (Kwakkel et al., 2015; Winstein et al., 2016). Therefore, one of the best current stroke motor rehabilitation therapies is only available to stroke survivors who have a mild impairment to begin with *or* stroke survivors who have already significantly recovered.

While recent successes in modelling, grouping, and predicting the trajectory of stroke recovery (Lundquist et al., 2021; Smith et al., 2019; Stinear et al., 2017a; van der Vliet et al., 2020) may be able to help us understand what is happening and how we can influence these processes, there is still a need for therapies that can allow stroke survivors to train consistently as early as possible after their stroke. This is a benefit that is innately available to Brain-Computer Interface (BCI) systems, as they can act as a bridge between brain and motor system. There are many different types BCI such as neuromuscular electrical stimulation, haptic and or robotic assistance, virtual or augmented reality responses as well as abstract feedback (Cervera et al., 2018) all of them have in common their ability to record, analyse and respond to brain signal without the need for input from the motor system as is the case for many conventional motor rehabilitation therapies.

Therefore, the healthcare burden caused by strokes could potentially be eased with rehabilitative BCIs. This is due to their unique ability to facilitate training in the early stage after stroke when little or no functional movement is possible for stroke survivors. Although BCIs have shown promising results in small-scale clinical trials (Nojima et al., 2022), their integration into broader clinical practice has been lacking (Baniqued et al., 2021). This thesis explores the underlying factors contributing to the limited clinical adoption and proposes a possible solution by introducing a fresh approach to BCI design, as well as a specific solution utilizing that approach. To set the stage, the next chapter will therefore summarise the landscape of medical BCI research.

More precisely, the next chapter entitled “Brain-Computer Interfaces in neurorehabilitation” and its corresponding publication entitled “Challenges and Opportunities for the Future of Brain-Computer Interface in Neurorehabilitation” addresses the question of why BCIs, despite encouraging results, are not widely used in stroke rehabilitation. Using a narrative review, we discuss recent meta-analyses of BCI for stroke rehabilitation while carving out the problems that medical BCI face and how they could be overcome. We found that scalability and medical justifiability are lacking in current BCI literature and argue that multimodal and multiphase BCI could present a solution by providing high validity answers to individual challenges. Therefore, the third chapter will investigate a possible multimodal and multiphase BCI approach.

In “Transcranial Magnetic Stimulation Neurofeedback – A Multimodal, Multiphase Approach to Stroke Rehabilitation Using BCI” (Chapter 3) and the corresponding manuscript “Optimising Control of Motor Imagery-based BCIs by Priming Using TMS-Neurofeedback”, we build on the suggestions of the second chapter. In this third chapter we designed an early poststroke rehabilitative BCI using a biofeedback paradigm called Transcranial Magnetic Stimulation Neurofeedback (TMS-NF) (Ruddy et al., 2018).

TMS-NF is a novel method to provide rich and muscle-specific feedback in a BCI (Ruddy et al., 2018). The TMS-NF setup consists of the patient sitting in a resting position while their hands are fitted with Electromyography (EMG) recording devices. One trial consists of one TMS pulse delivered over the motor cortex. The TMS produces muscle twitches in the participant’s hand called Motor Evoked Potentials (MEPs) the amplitude of which is recorded by the EMG-device. The participant is instructed to use Motor Imagery (MI) to increase or decrease the MEP amplitude in response to the TMS pulses. Therefore, a trial can have two conditions: “Up” where participants use MI to increase their response to the TMS pulse, or “Down” where participants use MI to inhibit their response to the TMS pulse. The MEP amplitude of a trial is compared to an average MEP amplitude measured before the TMS-NF. A successful Up trial is defined as a trial where the MEP amplitude is greater than the previously measured baseline. Conversely, a successful Down trial is defined as a trial where the MEP amplitude is lower than the previously measured baseline.

TMS-NF has several unique attributes that make it well suited to the purpose of poststroke motor rehabilitation. TMS-NF does not require actual movement, enabling training very early after stroke compared to conventional therapy, possibly taking advantage of synergies in the brief period of biological recovery. Unlike therapies such as robot assisted movement therapy and neuromuscular electrical stimulation therapy that do not require active participation, active participation is a key feature of TMS-NF. The participant attempts movement which encourages appropriate endogenous neural signal generation. This participation is then evaluated by the BCI, and feedback is provided for the participant to improve, hopefully strengthening the appropriate endogenous neural signal generation.

Another distinguishing feature of TMS-NF is that it intervenes by stimulating the brain but evaluates outcomes at the extremity, theoretically involving large parts of the motor pathway in the correct order. This design offers several advantages. Separating intervention and outcome location should improve signal to noise ratio at the outcome location enabling more accurate feedback. Further, engaging the motor pathway from brain to extremity should make it easier to correctly assess successful attempts resulting in more impactful training. It

is likely that there is a direct link between stroke recovery and TMS-NF as early after stroke the MEPs elicited from TMS pulses tend to be small and as the stroke survivors recovers motor function the MEP responses normalise (Stinear et al., 2015). Additionally, TMS-NF can be learned within two sessions (Ruddy et al., 2018) which is faster than many rehabilitative BCI, and TMS-NF has been shown to work in stroke survivors (Liang et al., 2020). However, TMS-NF requires trained administrators specialised equipment, making it unattractive for large-scale application.

Therefore, we decided to design a two-phase BCI where participants would first train with TMS-NF and then move on to train the same MI they learn during the TMS-NF with a more traditional EEG BCI. This has the advantage of combining TMS-NF – a modality that is quick to learn and well linked and justified for post stroke rehabilitation – with the potential scalability of an EEG BCI. This created a BCI system that uses two modalities (TMS and EEG) and has two distinct phases. We were interested to investigate how this multiphasic BCI design would compare against a monophasic design where participants learn the same MI instructions only using an EEG BCI.

We compared the performance of healthy participants using the multiphasic TMS-NF BCI with the performance of a monophasic group using only EEG BCI. We found that the multiphasic group learned to control the EEG BCI faster and more accurately than the monophasic group. Additionally, the multiphase group had different patterns of spatial and frequency activity making their motor imagery look more similar to processes of motor execution. The changed quality and quantity of training needed when using a multiphase BCI could prove very advantageous for rehabilitative BCI design. While this study and setup was complex, the multiphase approach can be simplified by testing individual phases and modalities, as we show in the fourth chapter.

In the fourth chapter “The Feasibility of Multiphase BCI Rehabilitation” and the according publication “A Wireless, Wearable Brain-Computer Interface for Neurorehabilitation at Home; A Feasibility Study” we investigate a potential third phase, where participants set up and train using an EEG BCI without supervision. For this we sent state of the art Bluetooth, dry electrode EEG headsets to healthy participants’ homes and instructed them to train using the same EEG BCI paradigm as used in chapter three. With minimal online instructions, participants were able to set up, train and achieve control over the BCI within six sessions. This opens the potential for a poststroke motor rehabilitation BCI pipeline, where stroke survivors can start with BCI therapy much earlier than traditional physical therapy and can have continued therapy at home with minimal supervision. The

successes of the studies in chapter three and four emphasise that simple building blocks used efficiently have the potential to substantially improve stroke rehabilitation. This runs counter to a current trend in BCI research that tends to use increasingly impressive technological solutions to chase increases in BCI control and accuracy.

The insights gained in chapter three and four are summarised and generalised in the fifth chapter “Contemporary Considerations for BCI Design in Neurorehabilitation” and the according publication “Keeping our Eyes on the Prize; Are we Losing Sight of the ‘Why’ in BCI for Neurorehabilitation?”. We explore how BCI research from the perspective of Bioengineers may differ from BCI research from the perspective of the medical community. We propose three guidelines to help focus the design of medical BCI: identifying the appropriate neural signal, identifying the therapeutic goal, and identifying the appropriate technology. The research presented so far has mainly dealt with motor rehabilitation after stroke, however, the principles behind BCI design, including the idea of multimodal and multiphasic BCI, presented in this chapter can be applied to other areas.

We therefore investigated other fields where BCI-based therapies could have an impact. In the sixth chapter “Applications of BCI for Rare Diseases and Disorders” and the according publication “Could Brain–Computer Interface Be a New Therapeutic Approach for Body Integrity Dysphoria?” we conceptualise design ideas for BCI or neurofeedback-based therapies for people suffering from Body Integrity Dysphoria (BID). We found that it is hard to imagine the consequences of such therapies, since little is known about the mechanisms of the disorder. However, since conventional pharmaceutical or psychological therapies have shown little to no effect, new ideas such as BCI-based approaches are welcomed. Further, BCI-based therapies for BID might benefit from the multimodal approach using Magnetic Resonance Imaging (MRI) as some implicated brain regions are located deeper in the brain and thus less accessible to EEG.

The last study of this thesis is presented in the seventh chapter “Structural Brain Connectivity in Elderly Individuals Relating to Motor Inhibition Networks” and the according publication “Stopping Networks and Falling in the Elderly: Relating White Matter Architecture in the Brain’s Inhibitory Control Networks to Real-World Metrics of Falling in Elderly Adults”. We investigate a research gap between real world behavioural data (falls) as a proxy measure for inhibitory control, and medical imaging (data). This research does not involve a BCI; It started as a smaller project to familiarise myself with the method of MRI which we planned to use for the final (unfinished) project associated with this programme of research, testing TMS-NF with stroke patients. Although not seamlessly

fitting within the BCI-related aims of this thesis, this MRI project can serve as an inspiration on how to design and assess the working mechanism of a potential future BCI that could increase inhibitory control by fortifying white matter structures and therefore reduce falls in elderly people.

In the final discussion of this thesis, chapter eight, I concisely summarise the main points of each chapter and discuss in more general terms the contributions they have made to the current literature within the broader field.

2. Brain-Computer Interfaces in Neurorehabilitation

2.1 Challenges and Opportunities for the Future of Brain-Computer Interface in Neurorehabilitation

2.1.1 Abstract

Brain-computer interfaces (BCIs) provide a unique technological solution to circumvent the damaged motor system. For neurorehabilitation, the BCI can be used to translate neural signals associated with movement intentions into tangible feedback for the patient, when they are unable to generate functional movement themselves. Clinical interest in BCI is growing rapidly, as it would facilitate rehabilitation to commence earlier following brain damage and provides options for patients who are unable to partake in traditional physical therapy. However, substantial challenges with existing BCI implementations have prevented its widespread adoption. Recent advances in knowledge and technology provide opportunities to facilitate a change, provided that researchers and clinicians using BCI agree on standardisation of guidelines for protocols and shared efforts to uncover mechanisms. We propose that addressing the speed and effectiveness of learning BCI control are priorities for the field, which may be improved by multimodal or multi-phase approaches harnessing more sensitive neuroimaging technologies in the early learning stages, before transitioning to more practical, mobile implementations. Clarification of the neural mechanisms that give rise to improvement in motor function is an essential next step towards justifying clinical use of BCI. In particular, quantifying the unknown contribution of nonmotor mechanisms to motor recovery calls for more stringent control conditions in experimental work. Here we provide a contemporary viewpoint on the factors impeding the scalability of BCI. Further, we provide an outlook for optimal design of the technology to best exploit its unique potential, and best practices for research and reporting of findings.

2.1.2 Introduction

Brain-computer interfaces (BCIs) are hailed as a promising approach to overcome paralysis by translating brain signals from movement intentions into computerised or motorised feedback. They can be used to restore, replace, enhance, supplement, or improve the natural output of the central nervous system (CNS), hence, providing opportunities for motor rehabilitation from a range of conditions including spinal cord injury, traumatic brain injury and stroke. Following a stroke, approximately 77% of survivors are left with some degree of upper limb impairment (Nakayama et al., 1994; Lawrence et al., 2001), which is a

key factor in preventing their engagement in normal activities of daily living and rendering them dependant on caregivers. Rehabilitation approaches that actively promote intensive and prolonged functional use of the paretic limb result in the largest gains in movement capability (Veerbeek et al., 2014). However, the gold standard approaches such as constraint-induced movement therapy (CIMT) require the patient to be capable of producing a sufficient level of functional movement in order to partake (Kwakkel et al., 2015). This prevents participation for those who are more severely impaired, or patients in the early weeks after brain injury who have not yet regained any function. With mounting recent evidence indicating that early intervention is crucial to harness the brain's endogenous recovery processes (Stinear et al., 2020), therapies that can support the patient to generate appropriate functional patterns of brain activity and motor behaviour are greatly needed, during the period when they are unable to generate actual movement unassisted.

2.1.3 Brain-Computer Interface for Neurorehabilitation; Basic Premise and Scope of the Review

Noninvasive brain-computer (and brain-machine) interfaces provide an advanced technological solution, decoding brain signals directly from the scalp and translating them into movement of a virtual (on screen) or robotic effector. The effector can also be the user's own limb, with electrical stimulation of muscles triggered by brain activity (Biasiucci et al., 2018; Bai et al., 2020). By closing the disrupted sensorimotor loop and providing tangible feedback, the patient learns to control the effector by motor imagery or movement intentions. Restoring relevant sensory feedback in relation to volitional movement attempts is believed to mobilise the fundamental mechanisms of motor learning (Mrachacz-Kersting et al., 2021). As such, they can engage in mental practise of movement and keep their motor neural circuitry active, warding off the detrimental effects of limb nonuse (Buxbaum et al., 2020), its associated white matter degeneration (Egorova et al., 2020), and promote use-dependant neuroplastic processes (Xing & Bai, 2020). Despite largely encouraging evidence suggesting that functional gains are produced exceeding those of standard care (Hattem et al., 2016; McConnell et al., 2017; Monge-Pereira et al., 2017; Cervera et al., 2018; López-Larraz et al., 2018; Raffin & Hummel, 2018; Carvalho et al., 2019; Coscia et al., 2019; Kovyazina et al., 2019; Remsik et al., 2016; Xing & Bai, 2020) substantial challenges with existing BCI implementations have prevented widespread adoption of the technology clinically (Baniqued et al., 2021). Here we provide a viewpoint on the practical, technical and mechanistic factors impeding the scalability of BCI into rehabilitative care packages. Further,

we provide a future outlook for optimal design of the technology to best exploit its unique potential, and best practices for research and reporting of findings.

Noninvasive BCI typically consists of three key components: A recorder, a decoder, and an effector. The recorder acquires brain signals from the scalp surface. The decoder analyses the recorded data, and the effector acts upon the information. In most cases the recorder is an electroencephalogram (EEG) detecting scalp electrical fluctuations associated with neuronal activity. In practise, any neural signal could be incorporated into a BCI, and implementations using magnetoencephalography (MEG) (Buch et al., 2008; Foldes et al., 2015), functional magnetic resonance imaging (fMRI) (Thibault et al., 2018) and functional near-infrared spectroscopy (fNIRS) (Soekadar et al., 2021) have also shown merit. The decoder is usually a programme run on a computer, extracting desired aspects from the signal and conducting analyses in real-time. The analysis process may be as simple as measuring amplitude or frequency of ongoing brain activity (Wierzgała et al., 2018), or more complex decompositions of inter-regional connectivity or dynamic changes to spatial patterns of activation (Rathee et al., 2017). The effector of a BCI can take multiple forms. For neurorehabilitation, it may be a device that assists the patient to complete movements, such as a robotic limb (Tariq et al., 2018; Soekadar et al., 2019; Khan et al., 2020), a device that gives virtual (e.g., on-screen) feedback to the participant to promote appropriate patterns of neural activity (Kerous et al., 2018; Si-Mohammed et al., 2018; de Castro-Cros et al., 2020), or a trigger to induce electrical stimulation of muscles in order to evoke movement (Biasiucci et al., 2018; Bai et al., 2020). Even in the absence of evoked movement, electrical stimulation can be used below motor threshold to provide continuous somatosensory feedback as the BCI signal (Corbet et al., 2018).

2.1.4 Practical and Technical Challenges With Clinical Implementation of BCI

For the BCI participant learning to control the effector requires multiple practise sessions viewing continuous feedback and learning by reward (Chavarriaga et al., 2017; Mrachacz-Kersting et al., 2021). While passive/implicit learning is known to play a role in BCI control (Othmer, 2009), most human participants report developing and fine-tuning mental strategies throughout the course of training, usually involving imagination of movement (Majid et al., 2015; Ruddy et al., 2018; Khan et al., 2020), or in the case of brain injured patients, attempts to make movement with the paretic limb (Blokland et al., 2012; Balasubramanian et al., 2018; Bai et al., 2020). Gaining useful control over an EEG BCI can

require a lot of training, especially if the BCI is designed to test medical hypotheses and not optimised for control (Pfurtscheller et al., 2003; Stieger et al., 2021). Without seeing tangible results within the first training sessions, it is likely that patients lose motivation to continue investing effort into trying to control the BCI. Another factor known to influence learning is the large variation in individual capability for motor imagery, contributing to the fact that 10–30% (Allison & Neuper, 2010; Vidaurre & Blankertz, 2010; Lotte & Jeunet, 2015) of users do not achieve control over the BCI; a phenomenon historically referred to as BCI illiteracy but more recently coined BCI inefficiency (Thompson, 2019). For BCI to serve as a useful therapeutic tool for neurorehabilitation, solutions that allow users to achieve control within a shorter time frame, and that are effective across a wider range of motor imagery capabilities, are needed to secure the future of this technology.

As the neural signals used to drive effectors in an EEG BCI are heavily influenced by ongoing mental state, the individual capacity to generate high quality mental imagery of movement dictates how easily detectable the relevant motor signals will be (Marchesotti et al., 2016; Chavarriaga et al., 2017). Visual motor imagery, imagining observing yourself perform a movement, produces less pronounced scalp-recorded signals than kinaesthetic motor imagery where the feeling of movement is effectively experienced (Neuper et al., 2005). Kinaesthetic motor imagery produces more easily detectable neural signals, at scalp locations overlaying motor cortical brain regions, and additionally modulates corticospinal excitability measured via motor evoked potentials (MEPs) in response to transcranial magnetic stimulation (TMS) (Stinear et al., 2006a). However in one study of 37 participants about half of participants found it difficult to perform kinaesthetic motor imagery (Seiler et al., 2017) and results of another study with 30 participants suggests that those would be the lowest performers on BCI (Vuckovic & Osuagwu, 2013). In some circumstances, it may be more beneficial to request that the patient make attempts to execute movements, rather than simply imagine movement (Blokland et al., 2012; Balasubramanian et al., 2018; Bai et al., 2020). For motor imagery-based BCI, multimodal or multi-phase BCI approaches (Leamy et al., 2011; Fazli et al., 2012) may lead to better accuracy, as different neuroimaging modalities may be more sensitive than EEG to detect very weak motor signals (albeit, technologies such as MRI, fNIRS, MEG, or TMS may be less practical for long-term practise of BCI). As an example of a potential multi-phase approach, using a BCI based not upon scalp signals but upon muscle responses to TMS over the motor cortex, control of on-screen feedback using motor imagery could be achieved within just two training sessions (Ruddy et al., 2018). Whereas EEG scalp signals associated with movement intentions have poor

spatial resolution, TMS can be used to target the motor cortical representations for specific muscles, selectively providing feedback of excitability of corticospinal pathways by measuring the amplitude of motor evoked potentials (MEPs) recorded at the muscle with electromyography (EMG). When tested in a sample of stroke patients, most were capable of learning to increase the excitability of their corticospinal pathways with TMS neurofeedback, using only motor imagery (Liang et al., 2020). Using a multimodal approach it may be possible to train individuals to develop robust motor imagery strategies that optimally excite the targeted brain-muscle pathways using TMS neurofeedback within just two sessions, that then translates to better subsequent performance using functionally relevant signals on an EEG BCI that has better portability options. This hypothesis remains to be tested empirically, and it is notable to point out that the approach may only be applicable to patients who exhibit MEPs when stimulated. Approximately 13.4% (Stinear et al., 2017a) are deemed “MEP negative,” and those tend to be the most severely affected (Stinear et al., 2017b; Smith et al., 2019; Lundquist et al., 2021). Incorporating multimodality into BCI paradigms may also extend beyond the aforementioned suggestions concerning acquisition modalities. Multimodal feedback (i.e., visual plus auditory or somatosensory) can also enhance the BCI learning experience (Sollfrank et al., 2016) and improve the quality of detectable brain signals (Sollfrank et al., 2015).

In order for BCI for neurorehabilitation to become scalable, it needs to answer to the current requirements of healthcare providers. Namely, it should reduce rather than increase burden and need for expert supervision, and instead place high quality rehabilitation into the hands and home of the patient in a cost effective manner. Even if initial control of the EEG BCI can be achieved more quickly using multimodal or multiphase approaches, longer term use over weeks or months would still be required alongside the patient’s standard rehabilitative care in order to promote functional upper limb improvement. Current implementations of EEG BCI are not well adapted for this purpose as they are cumbersome, require lengthy setup times with wet electrolyte-filled sensors, and a skilled operator to ensure sufficient signal quality, positioning of the headgear and execution of (often custom written and not user friendly) software. Recent technological advances in wireless, high impedance (dry electrode) EEG systems may enable better scalability. Using tablet-based software allowing real-time wireless streaming from a comfortable, wearable EEG cap with dry electrodes, signals of sufficient quality can be recorded even in home environments by elderly participants without assistance (Murphy et al., 2018a,b, 2019; McWilliams et al., 2021). Advancing this new technology to additionally provide real-time feedback to the

participant is a necessary next step towards home-based BCI that would allow extended training to be conducted in the weeks and months following brain injury. Further, it encourages the patient to feel in control of their own recovery process, rather than dependant on professionals or their caregivers. To date, existing implementations of wireless BCI for neurorehabilitation are at an early stage of technology readiness, with none reaching even small scale clinical trials (Baniqued et al., 2021).

2.1.5 Challenges for Elucidating Mechanisms Underlying BCI-Induced Functional Improvements

Advocating for clinical use of BCI is difficult when the specific mechanisms underlying functional improvements remain largely unknown. In order to make justifiable predictions regarding whether a patient is likely to benefit from BCI training, clinicians need to know what aspects of neural function are being targeted. The vast heterogeneity of available BCI methods further complicates attempts to elucidate mechanisms, as it is likely that different approaches target different aspects of neural circuitry to bring about functional improvements. A key issue to shed light upon across all types of BCI for neurorehabilitation is the potential contribution of nonspecific (i.e., nonmotor) mechanisms for recovery. Ros et al. (2020) name four nonspecific factors that may contribute to overall BCI performance improvement. These include (i) factors unique to the neurofeedback environment (such as trainer-participant interaction in a neurotechnology context). (ii) Factors that are common across interventions (such as benefits from engaging in a form of cognitive training, as well as psychosocial and placebo mechanisms related simply to participating in an experiment). (iii) Repetition-related effects and (iv) natural effects occurring during the intervention period such as cognitive development (in children/adolescents) or age-related cognitive decline in elderly participants. This list is, however, not exhaustive. Additional to the aforementioned mechanisms, BCI performance may also be influenced by effects from sustained exertion of effort or engagement, feelings of empowerment or competence from achieving control of the BCI, or general improvements in mood or enjoyment resulting from engaging with a challenging gamified task. Even with seemingly adequate control groups, it may be challenging to match aspects such as effort, attention, enthusiasm, and enjoyment between those receiving real neurofeedback and those receiving pseudofeedback which may be less intrinsically motivating. While it is encouraging that many studies are now including explicit measures to monitor training-induced changes to motor neural circuitry (i.e., using neuroimaging), measures of the aforementioned nonspecific effects are rarely included. Thus, their contribution cannot be evaluated with the currently available evidence.

While the presence of nonspecific BCI effects makes it more difficult to draw conclusions on how motor improvement for neurorehabilitation is achieved, it leaves open the intriguing possibility that although BCI training is conducted in the motor domain (i.e., using motor imagery), beneficial effects may not be exclusive to the motor system. For instance, it is conceivable that increased effortful focus on the BCI task over a sustained training period may lead to a top-down, generalised improvement in brain health, materialising as motor gains (measured by most studies) but also gains in other (e.g., cognitive) domains, which are not routinely quantified in BCI studies. This may materialise in the form of improved executive function, memory, attention, or processing speed. Such general improvements may be brought about by, for example, increased blood flow to the brain, enhanced neurochemical environment promoting plasticity induction, or simply increased traffic in neural circuitry sustaining healthy activity-induced myelination processes. Generalised (cross-domain) transfer from trained to untrained tasks is greatest when the trained task requires a high degree of attentional focus and cognitive flexibility (Green & Bavelier, 2008; Bavelier et al., 2018). In this regard, it is debatable whether the motor imagery BCI is primarily a motor task, or a cognitive task, making direction of transfer difficult to ascertain. Future work may focus on whether motor improvements arise as a result of transfer from improved cognitive function, or whether a portion of the specific motor improvements transfer to improve cognitive function. To test this empirically, design of future BCI studies should routinely include cognitive performance measurements alongside motor function tests, with measures taken at multiple timepoints throughout the learning process. Some tests that could routinely be used could include the Oxford Cognitive Screen which assesses multiple cognitive domains, the Hospital Anxiety and Depression Scale, and the Mental Imagery Questionnaire.

A small proportion of randomised controlled trials (RCTs) investigating BCI for neurorehabilitation have made efforts to measure and/or discuss potential mechanisms that give rise to functional improvements. Of these, candidate neural changes that co-occur with improvement in motor function include enhanced desynchronisation of sensorimotor rhythms (i.e., neural oscillations in the alpha 8–12 Hz and beta 13–30 Hz frequency range) over scalp locations corresponding to motor cortex (Buch et al., 2008; Prasad et al., 2010; Li et al., 2014; Ono et al., 2015), changes in functional connectivity (Varkuti et al., 2013; Pichiorri et al., 2015; Biasiucci et al., 2018; Wu et al., 2020), lateralisation of neural activity (Ramos-Murguialday et al., 2013) or changes to white matter microstructure (Song et al., 2015; Hong et al., 2017). Others have speculated that BCI works by mobilising the brain's intrinsic learning mechanisms, adapting behaviour using classical and/or operant

conditioning giving rise to neural adaptations (Mrachacz-Kersting et al., 2021). To date, there has not been a comprehensive account that successfully resolves the aforementioned different perspectives into a holistic mechanistic model encompassing the electrophysiological, haemodynamic, and neurochemical components. Multimodal investigations measuring BCI-related neural changes simultaneously in each of these different modalities (e.g., EEG, fMRI, and MR-spectroscopy) are warranted in order to understand how the neural elements interact to bring about functional motor improvements. A point to note is that in none of the above studies were nonmotor, nonspecific mechanisms tested, so their contribution to improving motor function remains unknown. Transfer of benefits to the nonmotor domain were also not quantified, leaving open the possibility that improvement in motor function may be a result of general brain health improvement.

Elucidating mechanisms of functional improvement from BCI is further complicated by the fact that brain injured patients have widely heterogeneous lesions, and lesion size and location do not predict functional outcomes in a straightforward manner (Umarova et al., 2021). Even in patients with similar extents of impairment, lesion location influences performance of BCI decoding of movement intentions (López-Larraz et al., 2017), and the scalp detected signals are qualitatively different when comparing cortical vs subcortical lesions in particular (López-Larraz et al., 2019). This poses challenges for a “one size fits all” approach to BCI for neurorehabilitation and stresses the importance of adaptive algorithms that do not make rigid a priori assumptions regarding location or characteristics of scalp detected signals, but rather allow flexibility to detect idiosyncratic patterns of neural activity that could be used to drive the BCI in an individually tailored fashion.

2.1.6 Outlook for Future Scalability and Justification of BCI Use Clinically

The field of BCI for neurorehabilitation has benefitted in recent years from collaborative efforts to standardise approaches using the best evidence-based technologies, and with recommendations for best practise in conducting research. For instance, the mental-task based BCI (MT-BCI) consortium is a multinational effort collecting the largest existing sample of BCI data across nine countries, with the objective to deepen understanding of learning mechanisms in MT-BCI, improve efficiency and reliability of MT-BCI and make them more useable for clinical and nonclinical applications (Jeunet et al., 2020). This “big data” approach to BCI breaks away from the typical small scale studies that are characteristic in this field, and may facilitate more advanced analyses techniques such as machine learning.

A key challenge is to make BCI tasks more user friendly, providing motivating feedback in a style that the user finds useful (Kübler et al., 2014; Pillette et al., 2017). Both hardware and software must be simple to use for patients and caregivers alike, which may result in greater enthusiasm towards the technology (Käthner et al., 2017). Tasks should avoid being fixed and repetitive, but rather should have an adaptive nature allowing the user to clearly see progression through stages as performance improves (Jeunet et al., 2016). BCI approaches that focus on assistance with activities of daily living (Soekadar et al., 2016) (particularly bimanual tasks in stroke patients) during physiotherapy may foster motivation and generalisation of skills towards everyday life (Soekadar et al., 2019). Additional to this, attempts to improve scientific rigour and reproducibility in neurofeedback research have established the CRED-nf framework for reporting of results, and recommendations for future design of studies (Ros et al., 2020). It is hoped that these collaborative efforts will improve understanding of BCI mechanisms by establishing a degree of standardisation of measurement, and ensuring that adequate experimental controls are in place.

2.1.7 Conclusions

The technological and practical scalability and clinical justifiability of BCI still pose challenges preventing widespread use for neurorehabilitation. Recent advances in knowledge and technology provide opportunities to facilitate a change, provided that researchers and clinicians using BCI agree on standardisation of guidelines for protocols and shared efforts to uncover mechanisms. Addressing BCI illiteracy and speed of learning are priorities for the field, which may be improved by multimodal or multi-phase BCI approaches harnessing more sensitive neuroimaging technologies in the early learning phases, before transitioning to more practical, mobile implementations. Clarification of the neural mechanisms that give rise to improvement in motor function is an essential next step towards justifying clinical use of BCI. In particular, quantifying the unknown contribution of nonmotor mechanisms to motor recovery remains elusive and calls for more stringent control conditions in experimental work. Measurement of additional neural (nonmotor) systems and of performance on nonmotor tasks is also essential to demonstrate specificity or transfer of the improvements across cognitive and motor domains. If the effects of motor imagery-based BCI are found to generalise beyond the motor system, for instance to improve cognitive control or gait, the potential relevance of BCI is expanded presenting an intriguing opportunity for the field. Ultimately, if the benefits are further found to generalise beyond lab-based experimental settings to more ecologically valid aspects affecting quality of life such as competence and autonomy (Lövdén et al., 2010; Cremen & Carson, 2017), the

intervention can truly be deemed as effective and worthwhile implementing clinically for neurorehabilitation.

2.1.8 Author Contributions

CS wrote the first draft and edited the final draft. KR and CS conceptualised the idea for the manuscript. KR, SS, NK, and DB worked on subsequent drafts of the manuscript editing and providing feedback. All authors reviewed the final version and were involved in discussions throughout.

CRedit Contributions

Conceptualization: C.S. and K.R.; Funding acquisition: K.R.; Investigation: C.S.; Methodology: C.S.; Project administration: C.S. and K.R.; Supervision: K.R.; Writing – original draft: C.S.; Writing - review & editing: C.S., D.B., N.K., S.S. and K.R.

2.1.9 Conflict of Interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

2.1.10 Acknowledgments

The authors would like to thank Alison Buick, Claire Donnellan, and Nicole Wenderoth for helpful comments and discussions relating to this work.

3. Transcranial Magnetic Stimulation Neurofeedback – A Multimodal, Multiphase Approach to Stroke Rehabilitation Using BCI

3.1 Chapter Introduction

Building upon the conclusions drawn in the previous chapter, we set out to design a Brain-Computer Interface (BCI) that could facilitate poststroke motor rehabilitation in the acute phase when patients have low or no movement capabilities but can produce Motor Evoked Potentials (MEPs). Stinear et al. (2015) have shown reduced corticospinal excitability in pathways connecting the primary motor cortex to the paretic muscles by measuring MEPs in response to Transcranial Magnetic Stimulation (TMS) in stroke survivors. They further showed that as the stroke survivors regain their motor function their corticospinal excitability increases proportionally. Ruddy et al. (2018) extended these findings to healthy participants and showed that a single pulse TMS protocol providing Neurofeedback (NF) in form of MEP amplitude allowed participants to volitionally increase or decrease their corticospinal excitability using different Motor Imagery (MI) strategies.

Given that we identified that BCI uptake and clinical use are impeded by the lengthy training times required for effective use, we aimed to test experimentally whether TMS-NF can facilitate faster learning to control an Electroencephalography (EEG) BCI. The primary hypothesis underlying our research question was that training with TMS-NF provides a richer, more muscle specific feedback than EEG BCI alone. Therefore, we compared learning rates between a group trained using TMS-NF in the early stages vs a group learning to control a traditional EEG BCI. Based upon previous work using TMS-NF with healthy adults (Ruddy et al., 2018) we chose a specific set of MI instructions that have been robustly shown to modulate corticospinal excitability, and whose neural mechanisms have been well documented.

3.2 Optimising Control of Motor Imagery-based BCIs by Priming Using TMS-Neurofeedback

3.2.1 Abstract

Currently, research for motor rehabilitation early after stroke is limited as this patient group is difficult to recruit efficiently and many rehabilitation interventions necessitate some

degree of functional movement. Paradoxically, as research suggests that there is a sensitive period of heightened brain plasticity after stroke, it is of paramount importance that motor rehabilitation should begin as early as possible. Motor Imagery (MI) based Brain-Computer Interfaces (BCI) can enable very early motor rehabilitation after stroke, but current Electroencephalography (EEG) based approaches require substantial time and effort to learn to use. We suggest that a multimodal and multiphase approach may allow stroke survivors to learn to use the BCI more rapidly, enabling early bedside intervention.

We designed a two-phase BCI for early poststroke motor rehabilitation. The BCI consists of a Transcranial Magnetic Stimulation (TMS) phase where participants learn to perform MI with the help of TMS-derived feedback of their muscle responses. We trained 17 healthy participants using TMS-Neurofeedback (TMS-NF) for two sessions, prior to conducting EEG BCI for three sessions. We compared the classification accuracy on the EEG BCI of the TMS-NF group to a control group learning for five sessions with only an EEG BCI.

We could confirm our preregistered hypothesis that the TMS group ($n = 17$) achieved better control and did so faster than the EEG group ($n = 17$). Further, the spatial distribution of imagery-based neural activity in the TMS group differed from the EEG group and more closely resembled patterns seen for movement execution. Lastly, alpha, beta and low gamma oscillations were correlated with BCI control.

The results indicate that priming using two days of TMS-NF may assist EEG-BCI users to develop robust mental imagery strategies that transfer well to an EEG BCI, enabling better control within a shorter timeframe. Given that EEG BCI is more portable and cost effective, this may enable a two phase approach where TMS-NF is used at the bedside in the early days after stroke followed by a longer period of self-directed EEG BCI rehabilitation that could occur in the home using wireless wearable technology.

3.2.2 Introduction

Stroke is already the second biggest contributor of disability adjusted life years, and current trends are likely to result increase total number of stroke survivors even with declining incidence (Feigin et al., 2021; Lawrence et al., 2001; Nakayama et al., 1994; United Nations et al., 2017; Virani et al., 2020; Vos et al., 2020). Advances in motor rehabilitation after stroke have, however, been limited within the last decade. More worryingly for the field, an emerging viewpoint is that rehabilitation interventions within the

first three to six months provide no benefits exceeding those observed with spontaneous proportional recovery.

According to the proportional recovery rule, the majority of stroke survivors regain around 70% of their potential total recovery within three to six months of the incident (Prabhakaran et al., 2008; Winters et al., 2015). The rule has been criticised (Hawe et al., 2019; Hope et al., 2019), but the strong correlation between the impairment immediately after the stroke and the impairment three to six months later has led pioneers in the field to conclude that “The rehabilitation received between these two time points is having no impact on recovery from impairment” (Krakauer & Carmichael, 2017, p. 69). According to Krakauer and Carmichael (2017), the conclusion is that the proportional recovery rule is capturing a spontaneous biological recovery process that is not affected by existing rehabilitation approaches. Since motor recovery after stroke progresses in a nonlinear, logarithmic fashion (Langhorne et al., 2011), bearing in mind that there is a phase of enhanced neural plasticity early after stroke (Krakauer & Carmichael, 2017; Stinear et al., 2020) the failure of conventional therapies to capitalize on this early window of enhanced biological recovery is a missed opportunity.

Recent successes in modelling, grouping, and predicting the trajectory of stroke recovery (Lundquist et al., 2021; Smith et al., 2019; Stinear et al., 2017a; van der Vliet et al., 2020) have contributed somewhat to our understanding of the recovery process and shed light on the value of early intervention. There is now a refreshed call for new therapeutic approaches that enable stroke survivors to train as early as possible after the incident. This is a quality that is uniquely characteristic of Brain-Computer Interface (BCI) systems that capitalise on harnessing neural activity associated with imagined movement, enabling very early training when little or no functional movement is possible.

The healthcare burden of stroke could potentially be eased with a new approach to BCI that allow patients to commence motor rehabilitation at the bedside using TMS-NF followed by an extended phase of at-home self-directed therapy using portable Electroencephalography (EEG) BCI (Simon & Ruddy, 2022). Although EEG BCIs have shown promising results in small-scale clinical trials (Nojima et al., 2022), their integration into broader clinical practice has been lacking (Baniqued et al., 2021). In order to maximise benefit and improve clinical acceptability, the BCI approach should evolve alongside changing needs of patients through different phases of recovery (Simon et al., 2021).

Neuroimaging has revealed that the neural activity associated with MI closely resembles that measured during actual executed movements (Héту et al., 2013), requires no

movement and can help with stroke recovery (López et al., 2019; Monteiro et al., 2021). While there are various different types of BCI that act upon different neural mechanisms, for stroke rehabilitation the goal is to provide tangible feedback of the otherwise hidden MI process in order to guide the patient to generate (and maintain) appropriate activity in their damaged motor circuitry. BCIs based upon scalp recorded EEG signals typically provide visual feedback of neural oscillatory activity within the alpha (8-15Hz) and Beta (15-29Hz) frequency range. Their spatial resolution is poor, and it is very challenging to differentiate patterns of activity associated with different imagined movements using the same limb. TMS-NF can be used to train individuals to selectively upregulate and downregulate the activity in corticospinal pathways connecting the brain to specific muscles, and as a result the EEG signals recorded simultaneously demonstrate distinctiveness even between imagination of different muscles on the same hand (Mihelj et al., 2021). Hence, it has been suggested that TMS-NF could be used as ‘priming’ to enable the development of robust mental imagery strategies that lead to distinguishable scalp-recorded neural signatures that could facilitate better control of an EEG-BCI.

In the current experiment we recruited healthy participants and tested whether two sessions using TMS-NF (on separate days) to learn to increase and decrease corticospinal excitability, as measured with Motor Evoked Potentials (MEPs), would improve subsequent performance and learning rate on an EEG BCI that relies upon classification of two distinct neural signals. We tested the following preregistered hypotheses (Ruddy & Simon, 2023):

- Hypothesis 1.1: MEP amplitudes in the TMS-NF group will be modulated by neurofeedback type (UP or DOWN). This is a one tailed directional hypothesis – predicted interaction between feedback type and block number.
- Hypothesis 2: Participants who are primed using TMS-NF for two days will demonstrate faster learning rates and more accurate control of an EEG BCI, compared to a control group. Directional-one tailed.
- Hypothesis 2.1: The group primed with two days of TMS-NF will start with better performance on their first EEG BCI session than the control group. Directional-one tailed.
- Hypothesis 2.2: By session 3 of EEG BCI, the TMS-NF primed group will perform better than the control group. Directional-one tailed.

- Hypothesis 2.3: At the end of EEG BCI training, the group who conducted their first two sessions with TMS-NF will perform better than the control group. Directional-one tailed.
- Hypothesis 2.4: The group primed with TMS-NF will learn to control the BCI faster than the control group.

And preregistered three exploratory hypotheses:

- Hypothesis 1.2 Participants can learn to modulate MEP amplitudes within two sessions (five blocks) even when the blocks are interleaved with UP and DOWN feedback types (i.e. replicating Ruddy et al. (2018) but in a shorter timeframe and more difficult task context. Directional-one tailed.
- Hypothesis 1.3: MEP modulation in the TMS-NF group will not be driven by increases in root mean squared Electromyography (EMG) activity. Directional-one tailed.
- Hypothesis 3: Participants primed with TMS-NF prior to EEG BCI training will develop more spatially distinct and temporally stable patterns of brain activity in motor regions, than the control group within the same timeframe. Directional-one tailed.

3.2.3 Methods

The study was approved by the Research Ethics Committee of Trinity College Dublin. All participants provided informed, written consent prior to participation. Participants chose between financial gratuity or research participation credits as a form of reimbursement. Healthy participants were recruited through convenience sampling on the campus of Trinity College Dublin using flyers, an online portal for study participation, or word of mouth. The inclusion criteria specified an age range between 18 and 60 years of age and right handedness. Exclusion criteria were as follows with consideration of the recommendations by Rossi et al. (2021): Any nonremovable and nonMagnetic Resonance Imaging (MRI) safe metal at or near the head, a history of seizures, any family history of seizures of direct relatives, a condition increasing the risk of seizure, a history of fainting, medication increasing the risk of seizure, intake of psychoactive or psychotropic substances, a neurodegenerative condition, neuropathy, and/or pregnancy. Appointments were scheduled at similar times, and participants were advised to keep to the same routine before appointments.

Participants

The sample size was based on a calculation performed using G*Power 3.1 (Faul et al., 2007), assuming a predicted effect size of $d = 0.97$ (MEP neurofeedback effect size from Ruddy et al., 2018), alpha value of 0.05, between two groups ($n = 17$ each). This test also assumes actual power of 0.95 for the resulting data.

From February 2022 to March 2023, 35 participants were recruited. One participant was excluded due to a technical error resulting in unusable data. The 34 participants (11 male) included in data analysis were all right handed (average score of 4.7 on Edinburgh five point scale (Veale, 2014)). The mean age of the participants was 21.66 years ($SD = 4.1$ years).

When registering for the study, participants completed a shortened version of the Kinesthetic and Visual Imagery Questionnaire (KVIQ) for MI ability focussing only the right upper body, consisting of a subscale for MI imagery and MI intensity (Malouin et al., 2007). Participants were allocated to a group using a randomisation and minimisation protocol to minimise differences in baseline MI ability between groups.

Data Collection Procedures

All participants took part in five experimental sessions on separate days (Figure 3.1). The TMS group ($n = 17$) underwent two sessions of TMS-NF, learning to upregulate and downregulate corticospinal excitability in their left First Dorsal Interosseus (FDI) muscle. The last TMS-NF session featured an additional shortened TMS-NF training session where EEG was recorded simultaneously. After the TMS-NF sessions they subsequently engaged in three sessions of a commonly used EEG-controlled BCI. The EEG (control) group ($n=17$) performed EEG BCI on all five sessions.

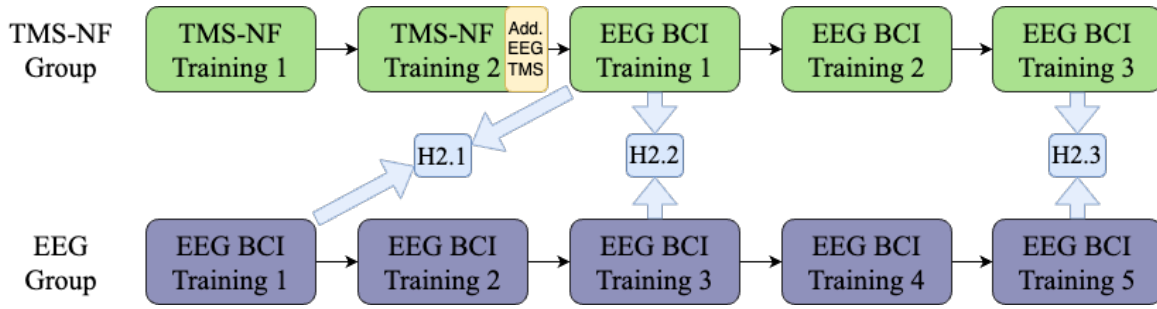


Figure 3.1. Timeline of the experiment. Each group trained a total of five times. The EEG group trained five times with the EEG BCI, and the TMS Group trained twice with the TMS-NF and then trained three times with the EEG BCI. All training sessions, EEG BCI and TMS-NF, trained with the same MI instructions. At the end of the second session there was a short break during which a TMS compatible EEG was set up and then participants trained for an additional half-session of TMS-NF with the EEG recording. The comparisons of hypotheses 2.1, 2.2, and 2.3 are highlighted. Hypotheses 2 and 2.4 were tested using a linear model and therefore do not use single time points for comparison.

TMS-NF Protocol

Participants in the TMS group underwent a two-phase training regimen with phase one being TMS-NF, described in this section, and phase two being EEG BCI, described in the following section. Phase one involved two TMS-NF training sessions separated by at least three days ($M = 4.13$ days, $SD = 1.49$ days). The goal was to train participants to use MI strategies to increase and decrease the size of MEPs in their FDI muscle by providing real-time visual feedback of their amplitudes in the form of a computer game. The setup has been previously described in (Ruddy et al., 2018).

The hands of the TMS-NF participants were prepared for electromyography (EMG) recording using a Biopac, Bionomadix, 2 channel wireless EMG transmitter and amplifier. Skin over the selected muscle locations was rubbed using an abrasive gel (Nuprep), and then sanitised with alcohol pads. Recording electrodes were placed on the FDI, the Abductor Digiti Minimi (ADM), and the three grounding electrodes on the knuckles corresponding to the fingers (index and little finger) and one on the outward facing wrist bone.

To find the motor hotspot (the location corresponding to the maximal excitation of the FDI muscle) participants wore a swimming cap upon which a 1x1 cm grid was drawn on the left side of the head. The origin of the grid was the middle of the participant's head, as measured from nasion to inion, and preauricular points. The participant was told to relax, look at the screen and think of nothing, while six MEPs were taken at the spot 5/1 on the grid (5 cm lateral, and 1 cm anterior from the origin), using a Deymed Duomag MP Stimulator with a Butterfly Coil (70BF). Discounting the first MEP (Pellegrini et al., 2018) the average taken at the spot was compared to the averages at the neighbouring spots. If any

spot had a higher average, the procedure was repeated at this new location. If no neighbouring spot had a higher average, the current spot was designated the motor hotspot and the administrator proceeded to take the resting Motor Threshold (rMT).

The rMT was found using a binary search until the lowest stimulator intensity output was found that could consistently elicit 5/10 MEPs above 25 microvolt. The rMT was then used as a baseline for the recruitment curve.

To build the recruitment curve participants received six TMS pulses at 90%, 100%, 110%, 120%, 130%, 140%, 150%, 160%, 170%, and 180% of the rMT in a random order. MEP amplitude was plotted for each intensity and the experimenter noted the intensity that produced MEP amplitudes that were half of the maximum recorded on the recruitment curve. This halfway stimulation intensity is thought to recruit approximately half of all muscle fibres available at this spot through TMS (Carson et al., 2013). This intensity was used throughout the remainder of the session, allowing participants equal scope to increase and decrease their MEP amplitudes using neurofeedback.

To establish a baseline MEP amplitude value to use as a benchmark for the training blocks, 20 additional TMS pulses at the halfway stimulation intensity were administered with the participant at rest, instructed to think of nothing in particular. These MEP amplitudes were averaged and served as a session baseline for the participant.

The TMS-NF training is depicted in figure 3.2 and described in detail in Ruddy et al. (2018), and summarised here. The gamified process consists of three different phases: the traffic lights, the fixation cross and the stimulation/neurofeedback phase. During the traffic lights phase the EMG activity of all four muscles is displayed in real-time to the participant in the format of four coloured ‘traffic lights’. If the EMG activity exceeds 3.5 microvolts the traffic lights turn from green to red. During the fixation phase a white cross is presented in the middle of the screen. Participants were instructed to begin MI during the fixation phase. During both the traffic lights and the fixation phases the trial will not proceed unless the EMG activity of all channels is below the 3.5 microvolt threshold for at least 0.5s. If participants consistently failed (after trying for several minutes and a relaxation break) to achieve below-threshold EMG activity for the TMS-NF, the threshold was increased in steps of 0.5 microvolt up to a maximum of 7 microvolts. After the fixation phase, a TMS pulse is administered and the participant’s MEP amplitude is displayed in real-time to the participant, accompanied by a negative or positive soundbite. The feedback consists of a white line, representing baseline recorded before the training, and a bar representing the current MEP amplitude compared to the baseline.

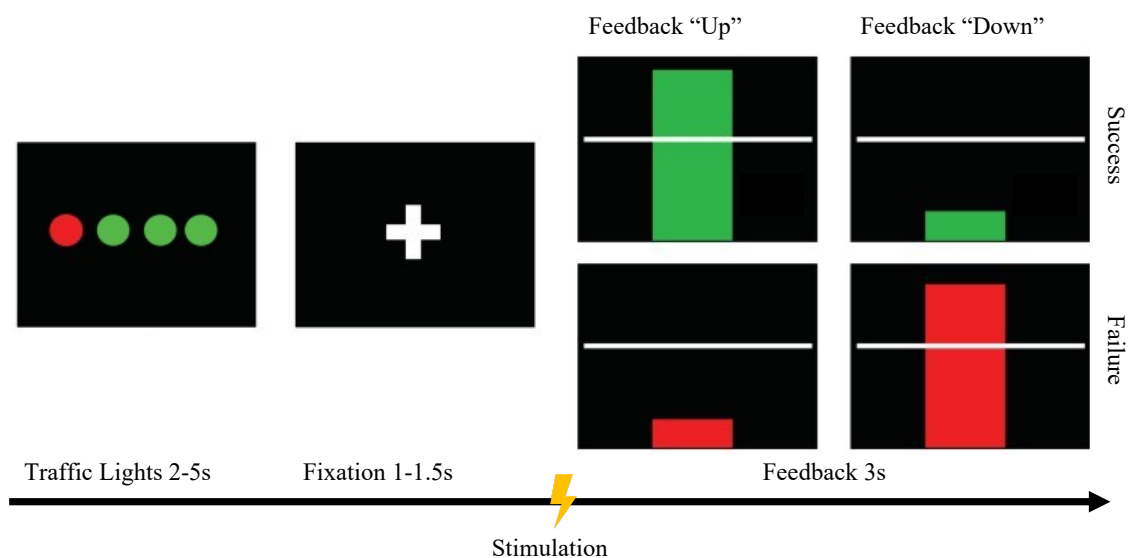
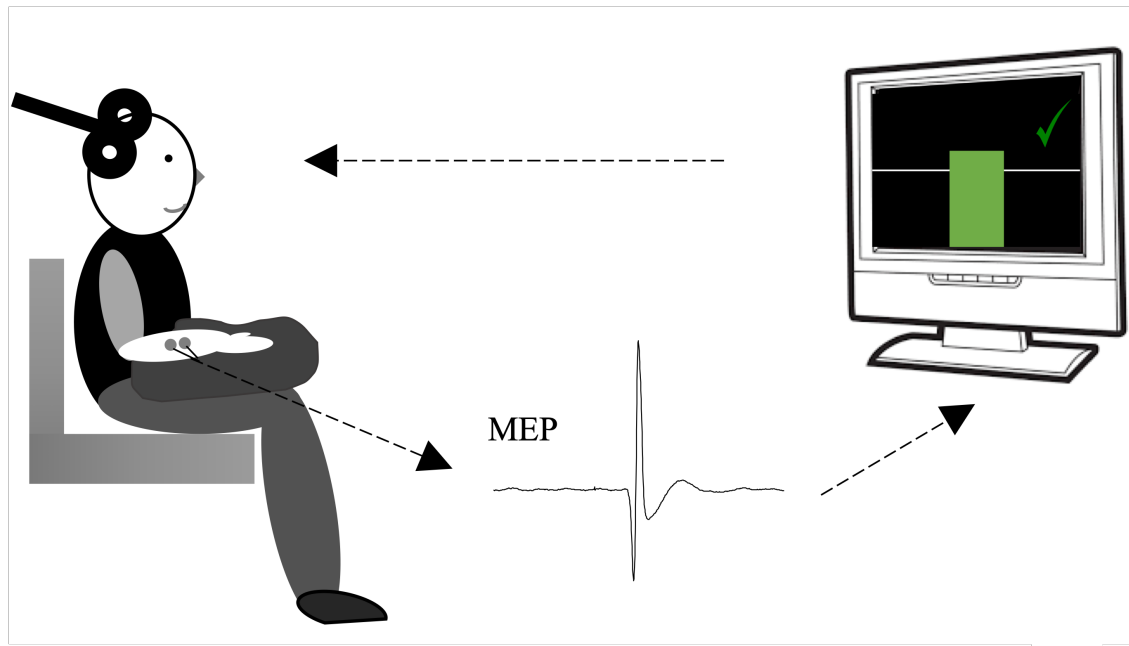


Figure 3.2. A short visual summary of TMS-NF. **Top:** TMS-NF setup. The participant sits and is at rest. The participant is asked to perform MI and a TMS pulse is triggered, resulting in an MEP. The MEP amplitude is displayed as feedback to the participant. **Bottom:** TMS-NF Protocol. The TMS-NF Protocol has three phases: the traffic lights, the fixation, and the feedback phase. The traffic lights turn red if the EMG activity of the participant surpasses a threshold. During both the traffic light and fixation phase the EMG activity is continuously monitored and progress to the next phase is blocked if it is above threshold.

The training consisted of two blocks of MEP amplitude upregulation (henceforth “Up”), and two blocks of MEP amplitude downregulation (henceforth “Down”). For the Up state, participants were asked to imagine complex or forceful movements of fingers on the right hand, which has been shown to upregulate corticospinal excitability of the pathways connecting the motor cortex to the muscles (Ruddy et al., 2018). For the Down state,

participants were asked to imagine that the right hand and fingers were cold, dead or detached, which has been shown to downregulate excitability in the same pathways (Ruddy et al., 2018; Majid et al., 2015). The training block conditions were randomised and counterbalanced within and between participants.

After the first four training blocks of the second session were over, the participants were prepared for the one more Up and one more Down block using a TMS-safe EEG device (Deymed TruScan LT 64). The preparation took around 40 minutes. Because of the break for the participant, the increased distance between TMS-coil, motor hotspot and the resulting increased stimulation intensity, it was necessary to retake the rMT. Therefore, for the analysis the EEG TMS part of the second session is separate from the first two.

After the training blocks were administered, we recorded 20 MEPs at the same stimulation intensity to assess any post-intervention changes.

After the two TMS-NF sessions, participants began phase two involving three training sessions with an EEG BCI on separate days.

EEG BCI Protocol

The EEG BCI protocol has been described in detail in Simon and Ruddy (2022) and will be summarised here. We adapted a Graz style BCI scenario in OpenViBE (Renard et al., 2010) to better suit the objectives of our experiment, and ultimately of stroke rehabilitation. Specifically, the traditional Graz BCI differentiates neural activity between imagined left and right hand movements using a Common Spatial Pattern (CSP) filter combined with a Linear Discriminant Analysis (LDA). The certainty of the BCI in classifying either left or right hand movement is the feedback displayed to the participant as a bar moving from right to left.

We specifically wanted to use the motor imagery strategies previously demonstrated to optimally up and downregulate corticospinal excitability, so we gave participants the same instruction as for the Up and Down states described previously in the TMS-NF protocol, and adapted the visual feedback to show a bar moving upwards (when neural activity was consistent with the Up state known to increase MEPs) and downwards (when consistent with the down state known to inhibit MEPs). The flow chart in figure 3.3 (taken from Simon & Ruddy, 2022) shows the task stages for the BCI online trials. The main outcome of the BCI was the Classification Accuracy (CA), defined as how accurately the BCI could discern which MI instruction (Up or Down) the participant was attempting.

The training consisted of five blocks of 30 trials, 15 trials per MI (15 Up and 15 Down). The trials were presented in randomised order. At the start of the first training block and after each training block, the participants rated their level of fatigue on the Karolinska Sleepiness Scale (KSS). The first training block had no online feedback and was used to calibrate the CSP and LDA. The remaining four blocks used the CSP and LDA to provide feedback to the participant.

A block of 30 trials was considered “balanced” if the BCI correctly recognised at least one trial of each condition. If the BCI categorised every trial of a block as only Up or only Down the block was classified as “unbalanced” – this was the case for 46 of the 568 blocks. After an unbalanced block the CSP and LDA were recalibrated with the data of the block before the unbalanced block. Then the unbalanced block was reanalysed using the updated CSP and LDA. The original, unbalanced, result was always kept to avoid inflating learning effects based on overfitted data. However, depending on the results of the new block, using the updated CSP and LDA, the administrator proceeded differently. If the new block was now balanced the administrator proceeded with the experiment using the new, updated, CSP and LDA. However, if the resulting block was still unbalanced, the administrator proceeded with the old CSP and LDA.

OpenViBE Processing

Before the CSP algorithm is used to maximise differences between the two MI, OpenViBE filters the incoming signal using a fifth order butterworth passband filter between 8-30Hz with a 1dB passband ripple.

For the LDA algorithm the raw signal is first filtered (using the same filter as described above), then the CSP is applied, and then every 0.0625s the signal is split into overlapping epochs of 1s duration. These epochs are squared, averaged and the logarithm is taken before the LDA algorithm is applied.

The online feedback processing is the same as for the LDA. The raw signal is first filtered (using the same filter as described for the CSP) before the CSP is applied, then the signal is split into epochs (the same as described for the LDA), and then these epochs are squared, averaged and taken the logarithm of before the LDA is applied.

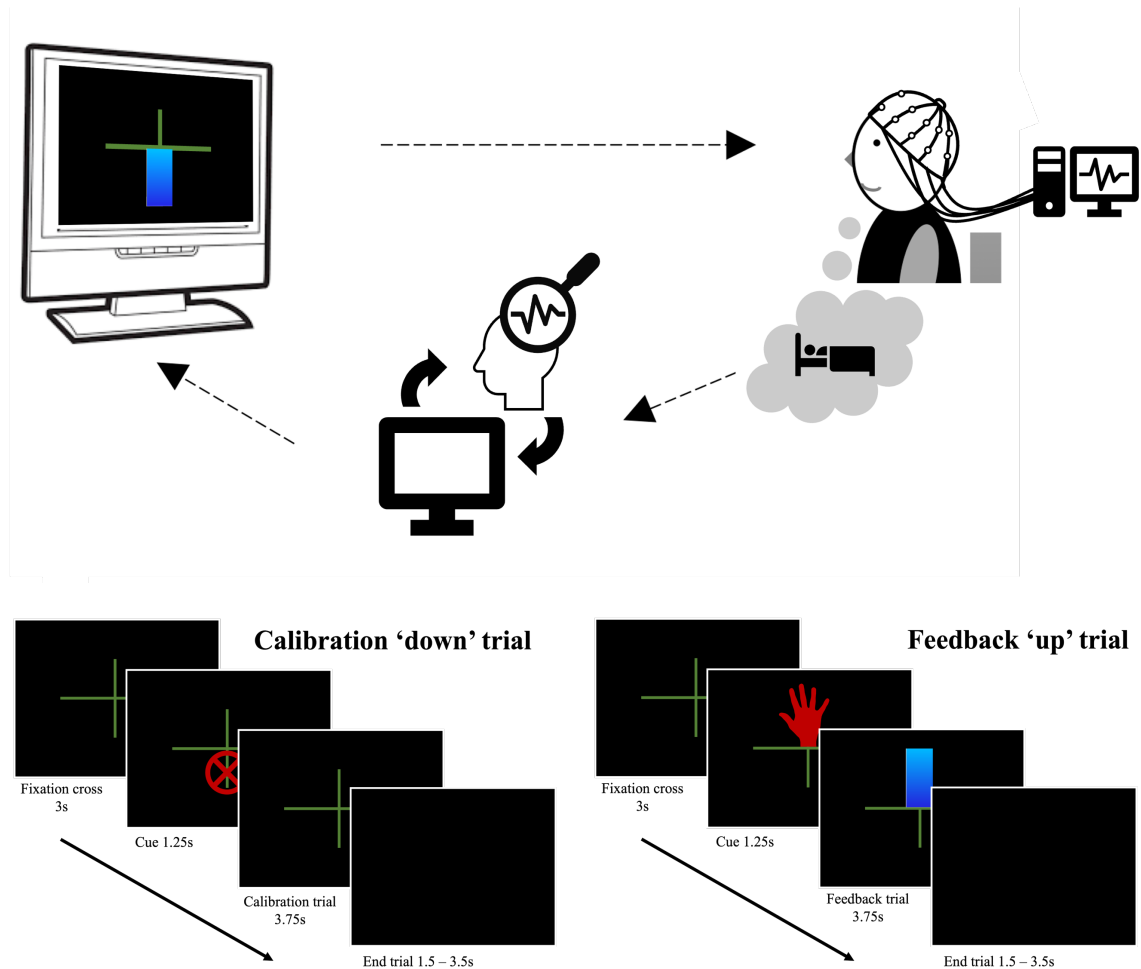


Figure 3.3. A short visual summary of the EEG BCI. **Top:** EEG BCI Setup. Participant sits and is at rest. A trial starts and the participant is asked to perform MI. The EEG records the brain activity. The EEG activity is classified by the computer in real-time, providing feedback to the participant. **Bottom:** EEG BCI Trial. Timeline of an EEG BCI trial for the calibration block (left, showing the cue for a Down trial) and online feedback blocks (right, showing the cue for an Up trial). Every session started with a calibration block without feedback to train the BCI and then proceeded with feedback trials. One complete trial was 9.5 - 11.5 seconds, and 30 trials made one block. Taken and adapted from Simon & Ruddy (2022).

Participants were fitted with a Biosemi Mark II 64 electrode system recording at 2048Hz, four additional electrodes were placed in the face, three accounting for Electrooculography (EOG) artifacts and one for jaw muscle activity. Electrode impedances were kept below 10 kOhm. The EEG signal was read in by an OpenViBE acquisition server. Each session started by recording one minute of blinking and one minute of horizontal and vertical eye movements to inform a fully automatic correction method for EOG artifacts (Schlögl et al., 2007). After this participants recorded three minutes of EEG signal at rest while fixating upon a cross. Participants then recorded a calibration block performing the required MI, used to train the initial CSP and LDA. These were fed into a real-time classification based BCI that displayed on-screen feedback showing in real-time what ‘state’

the classifier identified the ongoing brain activity as belonging to. If the instruction was to imagine hand movements (Up state) and the algorithm correctly classified the ongoing activity as belonging to this state, the bar was shown facing upwards, and the extent of the bar represented the confidence with which this classification was made. The refresh rate was 16Hz, giving the appearance of smooth real-time feedback of the neural activity associated with mental imagery.

Statistical Analysis

MEP preprocessing

The raw EMG epochs containing the MEP amplitudes were preprocessed for offline analysis. First the offset was removed from individual channels by averaging the period from 5ms to 100ms and subtracting it from the rest of the signal. Second the signal was highpass filtered (cutoff 30Hz) and lowpass filtered (cutoff 800Hz). To prevent artifacts due to the interaction between the TMS pulse and the filtering a subset of the epoch before the TMS pulse was taken and its data repeated. The original epoch was then filtered whole, and then the repetitions in the subset were discarded and the subset replaced the corresponding data in the original, filtered, epoch.

The background EMG activity before the pulse (105ms to 5ms before the pulse) was then calculated by obtaining the root mean square (Background RMS, or BrMS) for each epoch, and epochs with an average BrMS above ten microvolt were discarded. Epochs with highly variable data, defined as containing datapoints outside the range of the mean plus or minus 2.5 times the standard deviation of the epoch, were also discarded. All MEPs were visually inspected, and deviant MEPs were excluded from the analysis.

Classification Accuracy

We specified in the preregistration that, given fulfilled assumptions for the model, we would use a linear model similar to:

$$CA \sim group + time + group:time + (PID|PID)) \quad (1)$$

Where Classification Accuracy (*CA*) was the dependant variable and was predicted using the fixed effects called *group* and *time*, as well as an interaction effect between the two (*group:time*). *Group* stands for whether the value was recorded in the EEG or TMS group. *Time* stands for a variable denoting the cumulative amount of training the participant has done so far and assesses how training time affects the dependant variable. The model was

additionally enhanced with random effects according to individual participant intercept and slopes ($PID|PID$).

However, in practise the more complex model with random slope per participant still resulted in a fit that was not significantly better than a simpler model with no random slope and only a random intercept ($1|PID$) ($X^2= 1.89$, $p = .39$). Therefore, the simpler model was used (Equation 2):

$$CA \sim group + time + group:time + (1|PID) \quad (2)$$

MEP Amplitude

To estimate the effect of variables on the MEP amplitude, a variation of the model used for CA (Equation 1) was used. These models used MEP amplitude as a dependant variable (Equation 3) and did not have a group variable, as all MEPs were recorded in the TMS group. Instead, this model featured a variable called *condition* coding in which condition an MEP was recorded: Up, or Down. This model suffered from the same complexity issues as (1) and was simplified to include just one random effect for a random intercept per participant ($1|PID$):

$$MEP \text{ amplitude} \sim condition + time + condition:time + (1|PID) \quad (3)$$

t-Test decision tree

When directional t-Tests were performed, we first checked if assumptions for these tests were met. This meant checking both variables for normal distribution, visually and with a Shapiro test. We then performed an F-test to check for homoscedasticity. These tests informed us on whether to do a parametric (t-test) or nonparametric (Mann-Whitney / Wilcoxon Ranksum) test.

EEG Measures

For the EEG measures taken during BCI performance the trials were split into three epochs: trials of Up, trials of Down, and trials at rest (Figure 3.4). The rest trial was a period of two seconds before a cue appeared. This epoch was chosen to eliminate artifacts due to the participant “settling in” for the trial – as most movement artefacts occurred between the end of a trial and the cue appearing for the next one. For the Up and Down trials the epochs

were extracted from the middle of the motor imagery time window to avoid variability at the beginning or end of the trial (e.g. avoiding effects due to the cue disappearing at 1.25s).

EEG power timing

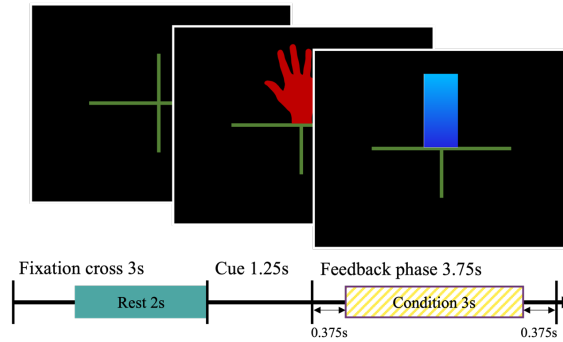


Figure 3.4. EEG Epochs. The epochs for the Rest condition began 2s prior to the appearance of the cue. The feedback conditions (Up or Down) were three seconds in the middle of the feedback phase, with 0.375s of buffer time before and after to avoid effects due to cue disappearance or anticipation of the end of the trial.

EEG Preprocessing:

The first preprocessing step was reconstructing the EOG corrected EEG data (see equation 4) described in Schlögl et al. (2007). The weights matrix produced by OpenViBE, representing the average activation associated with blinks and eye movement artefact data, was read into MATLAB (R2022a) and multiplied by the signal of the EOG channels (EOG signal). This was then subtracted from the raw EEG Signal (Consisting of all channels except EOG).

$$\text{EOG corrected EEG data} = \text{raw EEG signal} - \text{EOG signal} * \text{weights matrix} \quad (4)$$

To reduce bias in the EEG preprocessing, Artifact Subspace Reconstruction (ASR) was used for the preprocessing. ASR is a fully automatic preprocessing and data cleaning method (Gabard-Durnam et al., 2018; Mullen et al., 2013, 2015). The data were first downsampled to 250Hz, then a notch filter (48-52Hz) was applied to remove 50Hz noise associated with mains electricity. After this, a bandpass filter from 1 to 45Hz was applied. At this point, unused and nonEEG electrodes were removed, and the signal was averaged to

reference to provide a cleaner signal to the ASR algorithm. The ASR process used by EEGLab (version 2022.0, Delorme & Makeig, 2004) first rejected channels with more than five seconds of silence, more than four times standard deviation of noise, or a correlation of less than 0.8 with neighbouring channels. The ASR algorithm itself ran with default settings. These settings are described in detail in Simon and Ruddy (2022), and have been found to be conservative (Chang et al., 2018; Gabard-Durnam et al., 2018).

As described in Simon and Ruddy (2022), we then interpolated rejected channels back into the signal and performed another reference to average. We then ran an independent component analysis (ICA) with default parameters (Ablin et al., 2017). The ICA components were automatically labelled, and components with artefactual origin (Muscle, Heart, Line Noise or Channel Noise) with 90% certainty or more, were discarded. The input and output data for this automated process were visually inspected and detailed performance measures of the process were recorded. The last step of the preprocessing was applying a spherical spline surface Laplacian (Perrin et al., 1989) to increase localization of the EEG signal and reduce volumetric conductance effects.

EEG Power Metrics

After preprocessing we computed two different kinds of EEG power metrics: relative bandpower and percentage change from baseline. Both aim to simplify and standardise comparisons between powerbands. The power for each metric was calculated using Morlet Wavelets with 45 total frequencies ranging from 1 to 45Hz in 1Hz increments, with cycles increasing with frequency from 4 to 10.

Percentage Change from Baseline

Percentage change from Baseline was calculated using equation 5. Percentage change aims to show activity during a task by dividing each power band f at moment t during the task ($activity_{tf}$) by a mean baseline of the same frequency band ($\overline{baseline_f}$) recorded before the task (here the Rest condition). This circumvents the problem of the inverse relationship between amplitude and frequency of brain oscillations.

$$\text{Percentage change}_{tf} = \frac{activity_{tf} - \overline{baseline_f}}{\overline{baseline_f}} \quad (5)$$

However, percentage change is less sensitive than logarithmic transformations the further values are away from 0. Therefore, we additionally also computed the relative bandpower.

Relative Bandpower

The relative bandpower was computed using equation 6.2. and is a metric that compares the power of a powerband f at moment t ($activity_{tf}$) to the power of all other powerbands at moment t ($\sum_f dB_{tf}$). Therefore, to enable comparisons between powerbands the first computation step is a normalisation to tens of decibels by dividing the power of each powerband f at moment t during the task ($activity_{tf}$) by an averaged baseline of the same powerband ($\overline{baseline_f}$) recorded before the task (here the Rest condition) and taking the logarithm.

$$dB_{tf} = 10 * \log_{10} \left(\frac{activity_{tf}}{\overline{baseline_f}} \right) \quad (6.1)$$

$$\text{Relative bandpower}_{tf} = \left(\frac{dB_{tf}}{\sum_f dB_{tf}} \right) \quad (6.2)$$

Since logarithmic transformations are more sensitive to big changes than percentage change, relative bandpower may be able to pick up more nuances than percentage change. Inversely, it may also be more prone to noise.

Stability and Distinctiveness:

Stability and Distinctiveness have been described in Lotte and Jeunet (2017). We closely followed their implementation in Simon and Ruddy (2022) and again in this study. Both measures are implemented to address shortcomings of online CA measurements by conceptualising the variance within a MI state, and the difference between two MI states Lotte and Jeunet (2017).

We used the following implementation of stability (Lotte & Jeunet, 2017; Simon & Ruddy, 2022):

$$\text{Stability of Task A} = \frac{1}{1 + \sigma_{C^A}} \quad (7)$$

With σ_{C^A} being the standard deviation of the covariance matrix in Riemannian space. The covariance matrix being defined as the average of single trial (i) covariance matrices $C_i = \frac{1}{N_s} X_i^T X_i$ where N_s is the number of samples per trials, X_i is the EEG signal from all

channels, and T being transpose (Lotte & Jeunet, 2017). A full explanation of these concepts can be found in Lotte and Jeunet (2017).

We used the following implementation of distinctiveness (Lotte & Jeunet, 2017; Simon & Ruddy, 2022):

$$\frac{\delta(C^A, C^B)}{\frac{1}{2} (\sigma_{C^A} + \sigma_{C^B})} \quad (8)$$

Where the Riemannian distance $\delta(C^A, C^B)$ between the signals of two mental tasks A and B is divided by the averaged covariance matrixes $\left(\frac{1}{2} (\sigma_{C^A} + \sigma_{C^B})\right)$. In this case the two tasks, A and B, would be the two MI conditions of Up and Down.

Controllers

Controllers were defined as participants achieving a CA in their last session that is significantly greater than chance (50%) as determined with a t-test. This was the case for 10 of 34 participants (five from each group).

Common Spatial Patterns

CSP are a key feature of the classification approach relating to spatial distribution of the most influential signals driving the BCI. As such the data gathered about spatial distribution of activity patterns were also analysed. The 16 * 64 dimensional CSP files created by OpenViBE were averaged over the 16 dimensions to result in one weight per electrode. Those weight maps were separated by the four labels of controller, noncontroller, as well as TMS and EEG group.

3.2.4 Results

This section is subdivided by the three key measures of interest: the MEPs, the CA, and the EEG measures.

Motor Evoked Potentials

All results presented in this section concern the TMS Group and various influences on MEP amplitudes. According to the basic model for MEP amplitude described in equation 3 the interaction between training and MI condition is significant. This means that training with the MI instructions did systematically impact upon MEP amplitudes over time. Therefore, we can confirm hypothesis 1.1 and hypothesis 1.2. As can be seen in the figure 3.5 and table 3.1, the fixed effect for ‘Condition’ was statistically significant, which means that participants were able to regulate their MEP amplitudes using MI early on, without needing much training.

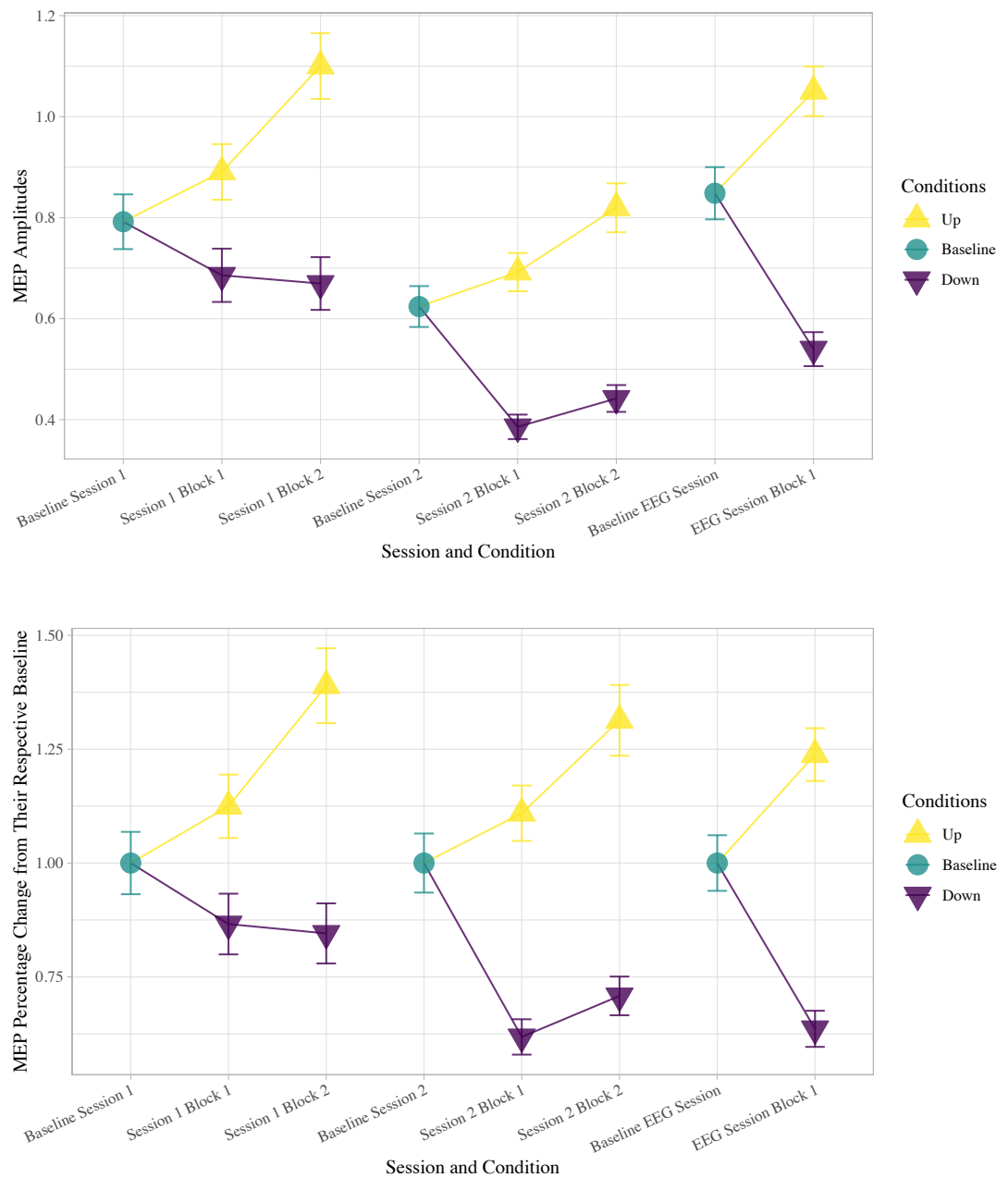


Figure 3.5. MEP Amplitude over Time. The top figure shows the MEP amplitude means from all MEP sessions. The bottom figure shows percentual change from the baseline (green) between the up- (yellow) and down-regulation (purple) condition. All comparisons between Up and Down are significant using the appropriate t-statistics and so is the effect of group in a mixed effects linear model predicting MEP amplitude or percentage change.

Table 3.1*Variables of the MEP Amplitude Model*

Variable	Beta	t-Value	p-Value
(Intercept)	0.56	6.8	0.001 >
Condition (Up)	0.17	4.16	0.001 >
Time	-0.0013	-2.76	0.0059
Condition (Up):Time	0.0026	3.67	0.001 >

Daily changes of MEP baselines are relevant

We investigated the negative effect of training time on the MEP amplitude. Instead of only accounting for time passed, we added a variable for different MEP sessions, accounting for changes in baseline. This means that the model could adjust for daily baseline MEP amplitudes and training time using separate variables. We also added another variable adjusting for the EEG TMS part of the second session.

The results in table 3.2 show that in the new model the effect of training time on the mean MEP values was attenuated. Instead, the beta value for the second session is negative while the TMS EEG part of the second session is positive. It is likely that participants felt less anxious about TMS in the second session leading to reduced MEP values (Oathes et al., 2008; Schutter et al., 2023). While the third session the length of the session and the combination of EEG and TMS may have again increased anxiety. The other terms of the model seem unaffected by this change.

Table 3.2*Variables of the MEP Amplitude Model, adjusted for different sessions*

Variable Name	Beta	t-Value	p-Value
(Intercept)	0.57	6.76	0.001 >
Condition (Up)	0.17	4.16	0.001 >
Time	0.00068	0.68	0.5
Second Session	-0.22	-5.1	0.001 >
EEG Session	0.12	3.18	0.0015
Condition (Up):Time	0.0026	3.75	0.001 >

A Chi-Square goodness of fit test was performed to determine whether one of the models was better at predicting the MEP amplitude. The model using two additional terms for the second session and the TMS EEG part of the second session was better: $\chi^2(2, N = 3211) = 65.4, p = 0.001 >$.

Background EMG activity is not responsible for the training effect

The link between BrMS and MEP amplitude is well documented in literature. Therefore, we must not only control for BrMS within the study design, but also account for it statistically.

Adding a term for BrMS and an interaction term between BrMS and the condition allows us to control if the BrMS is actually responsible for the training effect we observe. However, adding those two fixed effects does not change the significance of the training effect (Table 3.3).

Table 3.3

Variables of the MEP Amplitude Model, adjusted for Background Muscle Activity (BrMS)

Variable Name	Beta	t-Value	p-Value
(Intercept)	0.59	7.43	0.001 >
Condition (Up)	0.15	3.74	0.001 >
Time	-0.0012	-2.51	0.012
BrMS	0.21	8.12	0.001 >
Condition (Up):Time	0.0024	3.44	0.001 >
Condition (Up):BrMS	0.0045	0.15	0.88

Adjusting the model with the BrMS , reveals an association between BrMS and MEP amplitude. However, this model implies that the effect of MI on MEP amplitudes (*condition:time*, in table 3.3) is independent of BrMS. Therefore, preregistered hypothesis 1.3 can be confirmed. Additionally, the interaction between *condition*, training *time*, and BrMS is not significant. This means that the effect of training *time* and *condition* is not driven by BrMS.

We further investigated the link between BrMS and MEP amplitude by switching the model around; using BrMS as the dependant variable with the main effects of condition, time, the time and condition interaction, and an additional MEP amplitude term (Equation 9). We found no association between the BrMS and the condition, training time, or condition and training time (Table 3.4).

$$\text{BrMS} \sim \text{condition} * \text{time} + \text{MEP amplitude} (1|PID)) \quad (9)$$

Table 3.4*Variables of the model predicting Background Muscle Activity (BrMS)*

Variable Name	Beta	t-Value	p-Value
(Intercept)	-0.27	-2.52	0.021
Condition (Up)	0.06	1.53	0.13
Time	-0.00043	-0.9	0.37
MEP Amplitude	0.21	12.04	0.001 >

Dynamic thresholds may clean up noise

MEPs have high variability. A statistical method to eliminate variability due to different performance on different TMS sessions could be to adjust MEPs by their session average (dynamic thresholding). By doing so, raw MEP values are transformed into percentage values reflecting the percentage change from their session average.

Performing the analyses of Table 3.3, predicting CA using, time condition and BrMS, with dynamically thresholded MEPs instead of raw values strongly reduces the significance of BrMS in the model (Table 3.5). Further, the training time, condition and interaction are unaffected. This means some of BrMS' effect may be explained by the variability of MEPs from session to session.

Table 3.5*Variables of the MEP Amplitude Model, adjusted for Background Muscle Activity (BrMS)*

Variable Name	Beta	t-Value	p-Value
(Intercept)	-0.21	-5.08	0.001 >
Condition (Up)	0.19	5.05	0.001 >
Time	-6.3e-05	-0.14	0.89
BrMS	0.029	1.26	0.21
Condition (Up):Time	0.0024	3.74	0.001 >
Condition (Up):BrMS	0.045	1.64	0.1

Classification Accuracy Results

The CA results show that participants in the TMS group were able to learn to use the BCI faster and with better performance than the EEG only group, resulting in higher CAs. We can therefore confirm main hypothesis 2 using the preregistered analysis. In this model both the group term and the group and time interaction term significantly predicted CA:

Table 3.6*Classification Accuracy Model*

Variable Name	Beta	t-Value	p-Value
Intercept	62.61	17.15	0.001 >
Group (TMS)	-10.65	-2.1	0.038
Time	-0.16	-0.83	0.41
Group (TMS):Time	0.7	2.57	0.01

This model (Equation 2, Table 3.6) therefore confirmed that while participants in the TMS group started at a disadvantage of, on average, 10.65% lower CA, they gain 0.7 more with each training block than the EEG group. As the time variable ranges from 1 to 24, the model predicts that by the end of the experiment, TMS participants will have higher CA ($24 \times 0.7 = 11.76$ which is higher than the 10.65 penalty they incur).

The significance of the group by time interaction means that we can additionally confirm our preregistered main hypothesis 2.4 stating that the TMS group would achieve control faster than the EEG group.

Exploring the addition of covariates to the CA model

Adding a fixed effect for the motor imagery ability questionnaire (or KVIQ) score, does not change the outcome. The fixed effect itself is not significant, and not dependent on whether the total score or a subscore of the KVIQ questionnaire is used. Adding interaction terms makes a term significant: the interaction between the KVIQ subscale for MI imagery, group (TMS) and time (beta = -0.96, t-Value = -2.35, p-Value = 0.019).

However, adding a fixed effect for the sleepiness (KSS) score does increase the significance of the effects seen in the model (Table 3.7).

Table 3.7*Classification Accuracy Model, adjusted by KSS Scale*

Variable Name	Beta	t-Value	p-Value
Intercept	68.77	15	0.001 >
Group (TMS)	-12.11	-2.32	0.022
Time	-0.2	-1.0	0.32
KSS	-1.13	-2.23	0.026
Group (TMS):Time	0.78	2.77	0.006

This means that KSS has a significant impact on CA. To be certain that a difference in KSS scores was not the driving effect behind our training effect, it was important to add

a group, time and KSS score, and their individual and combined interaction with each other (Table 3.8).

Table 3.8

Classification Accuracy Model, Adjusted by KSS Scale and KSS interactions

Variable Name	Beta	t-Value	p-Value
Intercept	66.06	7.02	0.001 >
Group (TMS)	-26.54	-1.85	0.066
Time	-0.079	-0.12	0.91
KSS	-0.61	-0.366	0.72
Group (TMS):Time	1.95	2.0	0.044
Group (TMS):KSS	3.41	1.18	0.24
Time:KSS	-0.023	-0.18	0.86
Group (TMS):Time:KSS	-0.27	-1.36	0.17

As can be seen in table 3.8, adding interactions between KSS and the other independent variables does not change the significance of the group and time interaction. It is therefore a different effect. It is, however, weakening the significance of other effects, showing that sleepiness has an influence on BCI performance.

A Chi-Square Goodness of Fit Test was performed to compare the models. All models featuring KSS were equally better than the base model, of those the model using KSS without interactions of KSS was the simplest $X^2(1, N = 381) = 4.97, p = 0.026$.

Hypothesis 2.1: First EEG Session

Comparing the CA at the first EEG session between the EEG group ($M = 54.61, SD = 10.03$) and the TMS group ($M = 58.04, SD = 11.33$) there is a significant difference using a directional Welch two sample t-test $t(129.53) = 1.84, p = 0.034$, as described in the preregistration.

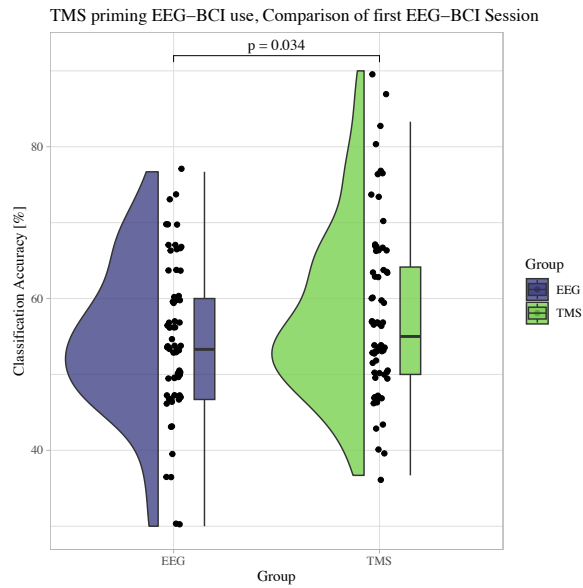


Figure 3.6. Comparison first EEG session. Distribution of classification accuracies for the EEG group Session 1 and the TMS group Session 3. This is the first time the respective groups use the EEG BCI. Using a directional Welch two-sample t-test, as described in the preregistration, the difference is significant.

This means that preregistered main hypothesis 2.1 can be confirmed, that TMS group participants have higher CA at their respective first EEG BCI session, as can be seen in figure 3.6.

Hypothesis 2.2: Third Training Session

This comparison is between the third total training sessions of both groups. The third training session for the TMS group is their first EEG BCI training session while the third training session for the EEG group is their third EEG BCI training session. The comparison of CA between the EEG group (median = 56.7) and the TMS group (median 55) is not significant using a directional Mann Whitney U test with continuity correction as described in the preregistration $W = 2334$, $p = 0.77$.

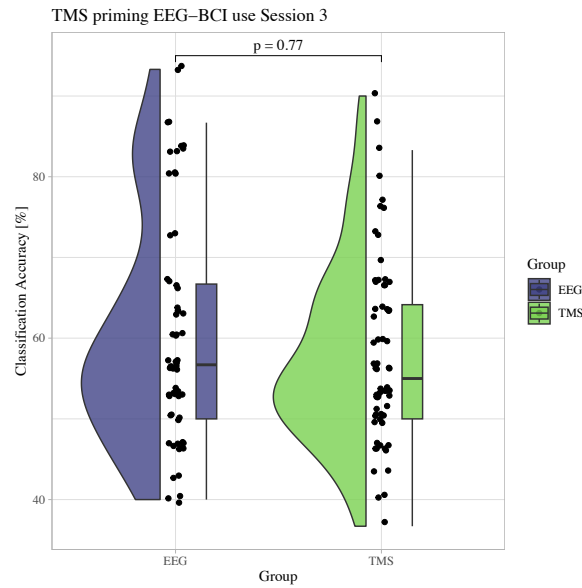


Figure 3.7. Comparison session 3. Distribution of classification accuracies for the EEG group Session 3 and the TMS group Session 3. This is the third time the EEG group uses the EEG BCI, and the first time for the TMS group. Wilcoxon rank sum test with continuity correction as described in the preregistration, the difference is not significant.

This means that preregistered hypothesis 2.2 cannot be confirmed; TMS group participants do not have higher CA at the third session, as can be seen in figure 3.7.

Hypothesis 2.3: Last EEG Session

Comparing the CA in the last EEG session between the EEG group (median = 55) and the TMS group (median = 60) there is a significant difference using a directional Wilcoxon rank sum test with continuity correction as described in the preregistration $W = 1777.5$, $p = 0.034$.

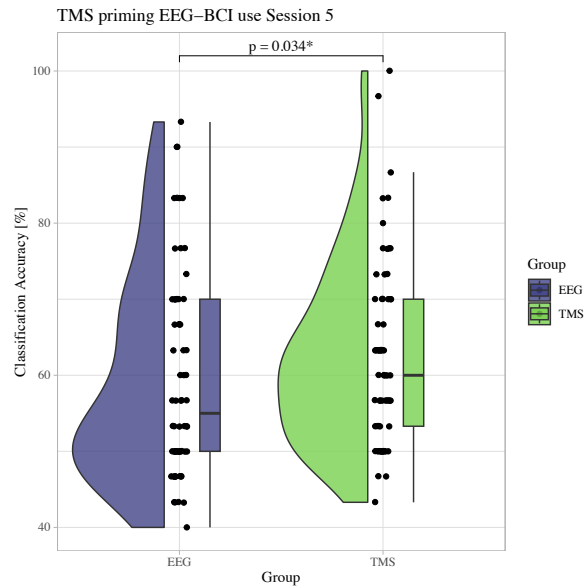


Figure 3.8. Comparison session 5. Distribution of classification accuracies for the EEG and TMS group Session 5. This is the fifth time the EEG group uses the EEG BCI, and the third time for the TMS group. Wilcoxon rank sum test with continuity correction as described in the preregistration, the difference is significant.

This means that preregistered hypothesis 2.3, that TMS group participants have higher CA at the end of the training, can be confirmed, as can be seen in figure 3.8.

Explorative Analyses

Instead of looking at the agglomerative measure of CA, we also looked at individual results for Up and Down MI. For this we used the model presented in equation 4, but substituted CA for successes in Up or Down MI.

The model where CA was substituted by successes for Down MI has no significant variables.

The model where CA was substituted by successes for Up MI has two significant variables. The variable for time is significant (Beta = 0.087, t-Value = 2.16, p-Value = 0.03). The interaction term between the group and time is significant (Beta = 0.21, t-Value = 2.28, p-Value = 0.023). It is also interesting to note that in this model, the time variable represents training itself. Meaning that being part of the TMS group more than doubles the effect of training from an increase in CA of 0.07 per unit trained to an increase of 0.21.

EEG Measures

We looked at several different measures of EEG activity. This part of the results is split into four subcategories: Stability, Distinctiveness, CSPs, and EEG Power.

Stability

Stability of the MI patterns were not significantly influenced by being in the TMS or EEG group, nor training time or the interaction between the two. However, it is strongly negatively affected by the KSS score for all MI patterns. Table 3.9 shows the results of a model predicting the stability of Rest, Up or Down using a variable for time, group and their interaction term. Therefore, preregistered hypothesis 3 is partially not confirmed.

Table 3.9

Model Predicting the Stability of Rest

Variable Name	Beta	t-Value	p-Value
Intercept	0.16	33.52	0.001 >
Group (TMS)	0.00093	0.17	0.87
Time	-0.00012	-0.63	0.52
KSS	-0.0022	-4.27	0.001 >
Group (TMS):Time	-0.00029	-1.02	0.31

Distinctiveness

Distinctiveness of the neural activity patterns comparing Rest to Down or Rest to Up is not significantly influenced by being in the TMS or EEG group nor training time nor the interaction between the two.

However, when looking at the distribution of the distinctiveness between Up and Down in a density plot (Figure 3.9) it is clear that there is a bi/multimodal distribution, which is statistically confirmed by an excess mass (0.11) test, p-Value < 0.001.

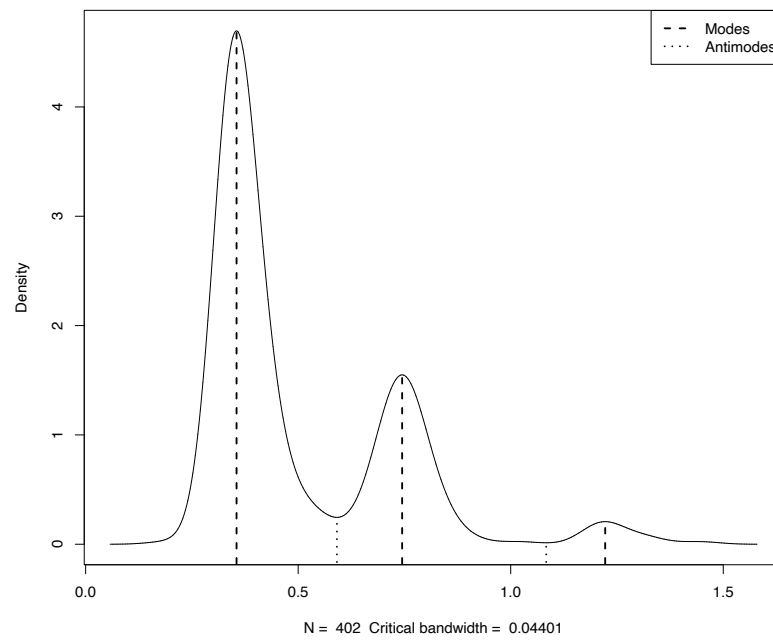


Figure 3.9. Bimodal Distribution of Distinctiveness Up vs Down. Distribution density plot of the distinctiveness between the Up and Down MI states of all participants. It can clearly be seen that there are two distributions, one with a peak at 0.36 and one at 0.74.

The cause of this bimodal distribution was not found. We tested many different variables and did not find a correlation between them. For modelling purposes we used both the complete and the individual distributions.

We did find that using the “first” distribution centred around 0.36 (separation done at the antimode of 0.59) yielded a significant result for predicting CA using the distinctiveness of Up vs Down, the results are summarised in table 3.10.

Table 3.10

Model Predicting CA Using the Distinctiveness Between the MI State of Up and Down

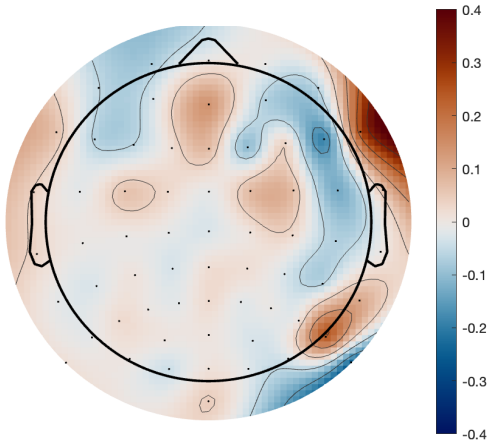
Variable Name	Beta	t-Value	p-Value
Intercept	49.83	7.45	0.001 >
Group (TMS)	0.52	0.05	0.96
Time	- 0.17	0.82	0.41
Distinctiveness Up v Down	30.4	2.04	0.042
Group (TMS):Time	0.79	2.59	0.01
Group (TMS):Distinctiveness Up v Down	-30.6	- 1.44	0.15

Common Spatial Patterns

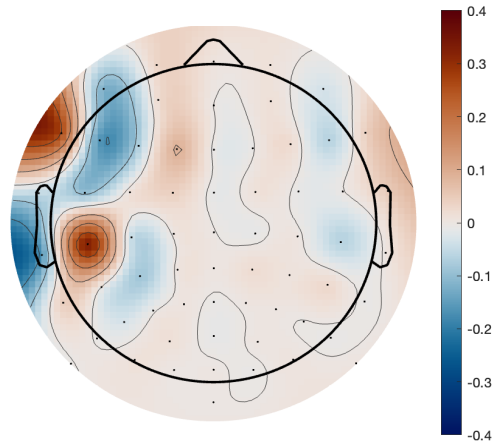
The pattern of CSP weights was not equally distributed between the groups of TMS and EEG users as seen in figure 3.10. To investigate this phenomenon participants were split into four groups of TMS or EEG controllers and noncontrollers respectively. EEG signal within these groups were concatenated, and a new CSP weight distribution with 16 features was calculated per group based on the agglomerated signal. The top and bottom four features of this CSP matrices were then taken and averaged, representing the most important features of the Down and Up conditions. These data were then represented in a topoplot (Figure 3.10)

In figure 3.10 it is visible that the EEG group (left side) uses a frontal and right lateralised pattern of activity for the Down condition and a narrow band arranged anterior to posterior over the centre for the Up condition. However, the TMS group uses a strongly left lateralised weight distribution for the Down condition, and a larger, left to right band of weights over the centre for the Up condition. This means that visually, hypothesis 3 can be confirmed.

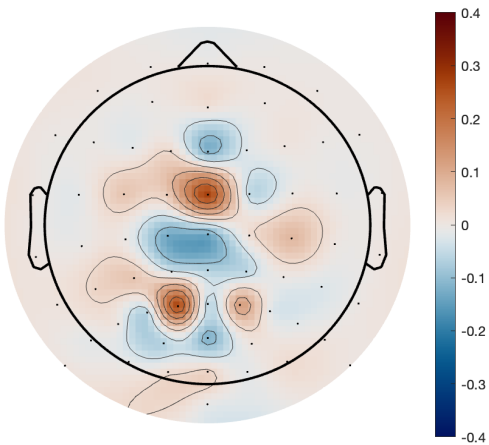
A. EEG Controllers Down



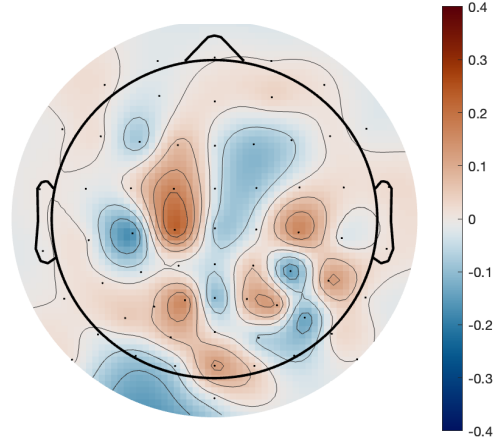
B. TMS Controllers Down



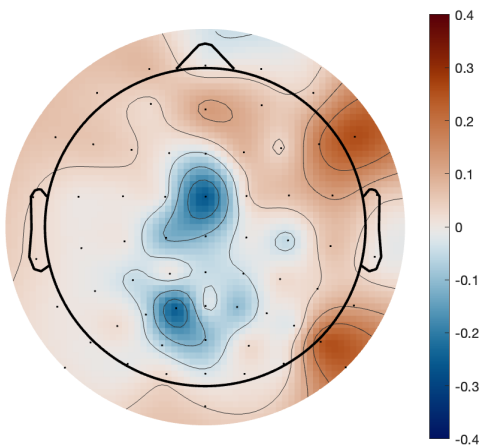
C. EEG Controllers Up



D. TMS Controllers Up



E. Difference EEG Down – Up



F. Difference TMS Down – Up

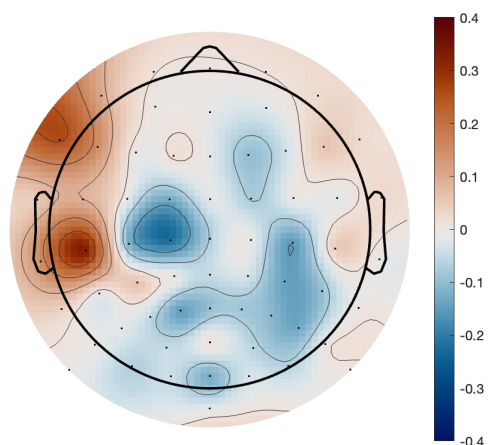


Figure 3.10. CSP weight distributions. **A-D:** These topoplots depict the averaged weights of the top or bottom four features of the CSP matrix differentiating between Down and Up conditions. **A.** No clear pattern is recognisable. **B.** Weights are left lateralised, possible hotspot in C5. **C.** Central pattern recognisable with dipoles arranged anterior to posterior, possible hotspots at P1 and FCz. **D.** Central pattern recognisable, with dipoles arranged medial to lateral, possible hotspots at C3 and C1. **E-F:** These summary topoplots of the CSP

weights take the absolute weights per group of the Down condition (A and B) and subtract the absolute weights of the Up condition (C and D). This means they show the preferential weight distribution of the CSPs between the Down, in red, and Up, in blue, conditions, per group. It is visible that both groups had central distributions for the Up condition (blue), which seems more left lateralised in the TMS group. However, the groups CSP weights distribution differ for the Down condition, where the TMS group has a strongly left lateralised pattern, while the EEG group shows a right lateralised pattern.

Using a Mann Whitney U test the difference between the weights of the clusters is significant for the TMS cluster when comparing EEG (median = 0.017) and TMS controllers (median = 0.026) $W = 263$, $p = 0.046$. None of the other comparison, that is, comparing each group on the TMS cluster between noncontrollers, and EEG clusters between controllers and noncontrollers is significant.

EEG Power

To investigate the EEG power, we analysed the power of a cluster of electrodes cluster CP3, C3, CP1. These electrodes were chosen before the CSP analysis was completed, based on a preliminary analysis showing that these were highly weighted in the TMS group. The pre-processed data into seven different power bands.

Table 3.11

Powerband Split

Power Band Name	Minimum	Maximum
High Delta	1	5
Theta	5	8
Low Alpha	8	11
High Alpha	11	14
Low Beta	14	22
High Beta	22	31
Low Gamma	31	45

Note: The lowest frequency of any given band was equal to or higher than its minimum, and lower than its maximum.

We tested each power band percentage change and relative band power for its predictive power for CA, Up and Down. Therefore, each model was tested six times using the formula:

$$CA = group * time * power\ band + (1|PID) \quad (10)$$

This means every model included four power band terms: a main effect of the power band itself, *group*, and *time*, an interaction between the power band and *time*, an interaction with power band and *group*, as well as an interaction between power band, *group* and *time*. A summary of the results is in table 3.12. We have categorised the results into five different patterns, expressing how closely associated with the results they were: from Category 1 where a powerband was significantly associated with the results of CA, Up and Down, to Category 5 where no significant association was found. In order to present concise results we have picked models we think represent the data in each category. However, all model data is available in the appendix.

Table 3.12

Summarised Results of Powerbands

Power Band	Percentage Change			Results Pattern	Relative Power			Results Pattern
	CA	Up	Down		CA	Up	Down	
High Delta	-	*	*	2	-	*	*	2
Theta	*	-	-	3	-	-	-	5
Low Alpha	-	*	*	2	*	*	*	1
High Alpha	-	*	*	2	*	*	*	1
Low Beta	-	-	*	4	-	*	*	2
High Beta	-	-	-	5	*	-	-	3
Low Gamma	-	-	-	5	*	*	*	1

Note: A “-” signifies no significant result. A “*” signifies any significant result between of the Powerband terms. According to the pattern of results, we grouped the results into five different result categories. Category 1 indicates that the powerband was significantly associated to all results (CA, Up and Down). Category 2 indicates that the powerband was significantly associated to the results of the Up and Down states, but not significantly associated to the results of CA. Category 3 indicates that the powerband was significantly associated to the results of CA, but not significantly associated to the results of either Up or Down. Category 4 indicates that the powerband was significantly associated to the results of either Up or Down, but not significantly associated to the results of CA. Category 5 indicates that the powerband was not significantly associated with any outcome.

Results Pattern 1

All results pattern of category 1 were significantly associated with the results of CA, Up and Down. This was the case for the relative bandpower powerbands of low alpha, high alpha, and low gamma. The results for high alpha and CA are presented in table 3.13, figure 3.11, representatively for all Category 1 results. The same figures and tables are available for Low Alpha and Low Gamma in the appendix.

Table 3.13*Model Predicting CA Using High Alpha (11-14Hz) Relative Power*

	Beta	t-Value	p-Value
(Intercept)	164.54	5.22	0.001 >
Group (TMS)	-95.71	-2.28	0.023
Time	-7.3	-3.62	0.001 >
High Alpha	-42.73	-3.27	0.0011
Group (TMS):Time	6.69	2.48	0.014
Group (TMS): High Alpha	35.79	2.04	0.041
Time: High Alpha	2.99	3.56	0.001 >
Group (TMS):Time: High Alpha	-2.51	-2.22	0.027

All variables are significantly associated with CA. The low alpha relative bandpower main effect is negative (-42.73), and so is the interaction effect of low alpha relative bandpower on the TMS group over time (-2.51). However, the high alpha relative bandpower and time interaction is positive (2.99), and so is the effect on the TMS group (35.79). Visually, this is represented in figure 3.11.

Association Between Relative Power, High Alpha 11–14 Hz Power And CA Split by Session, Control and Group

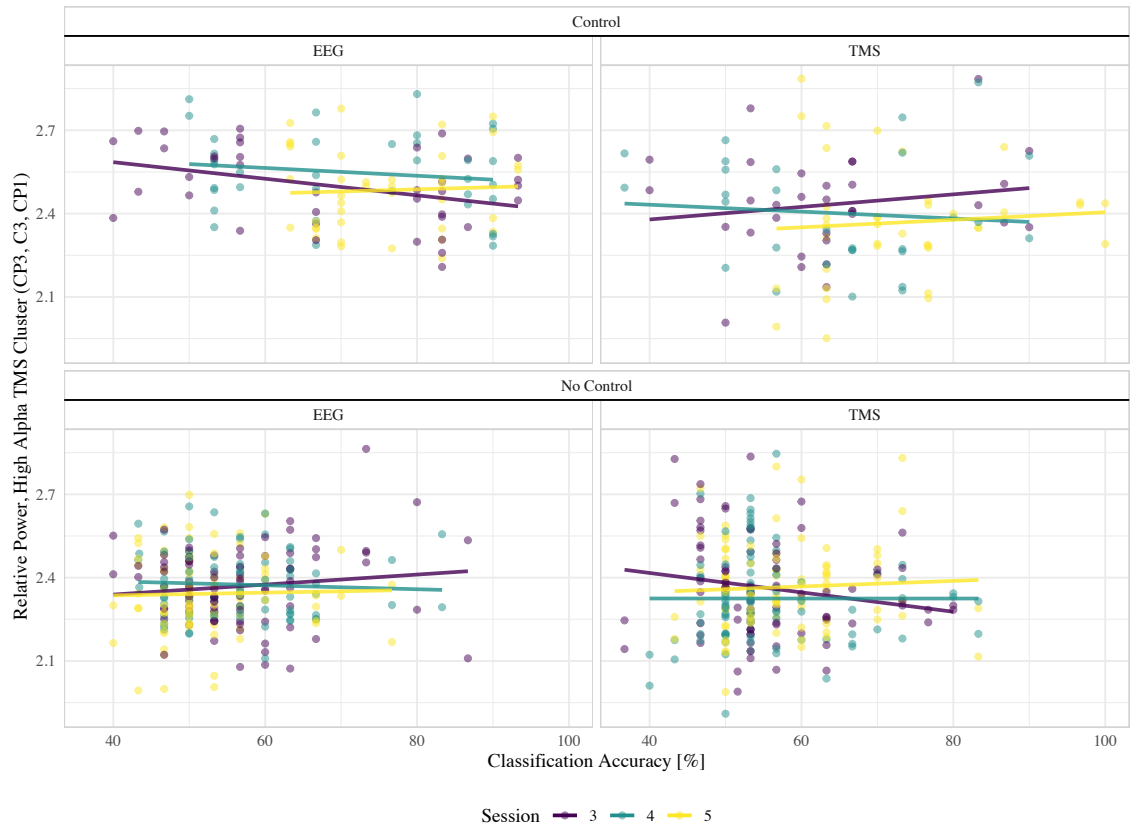


Figure 3.11. Relative Power High Alpha Split. This figure shows the low alpha mean power on the y axis and classification accuracy on the x axis. The plots are split by group with EEG in the left and TMS in the right column. The plots are also split by control achievers (top row) and noncontrollers (bottom row). Lastly, The different sessions are coloured. There is no easily discernible pattern.

Figure 3.11 does not show a clear pattern. A clearer pattern emerges when we further split up the data into Up and Down conditions (Figure 3.12). TMS controllers end with clear differences in relative bandpower high alpha (Up: $M = 2.47$, $SD = 0.26$; Down: $M = 2.32$, $SD = 0.2$, $t(19) = 2.98$, $p = 0.0078$, using a paired t-test. Further, there is a difference between the EEG Down condition ($M = 2.46$, $SD = 0.13$), and the TMS Down condition ($M = 2.32$, $SD = 0.2$), $t(38) = 2.7$, $p = 0.01$ using an independent t-test. Visually, this difference may already be present at session 3.

Relative Power, TMS vs EEG Group, Comparison of Third Session vs Last Session

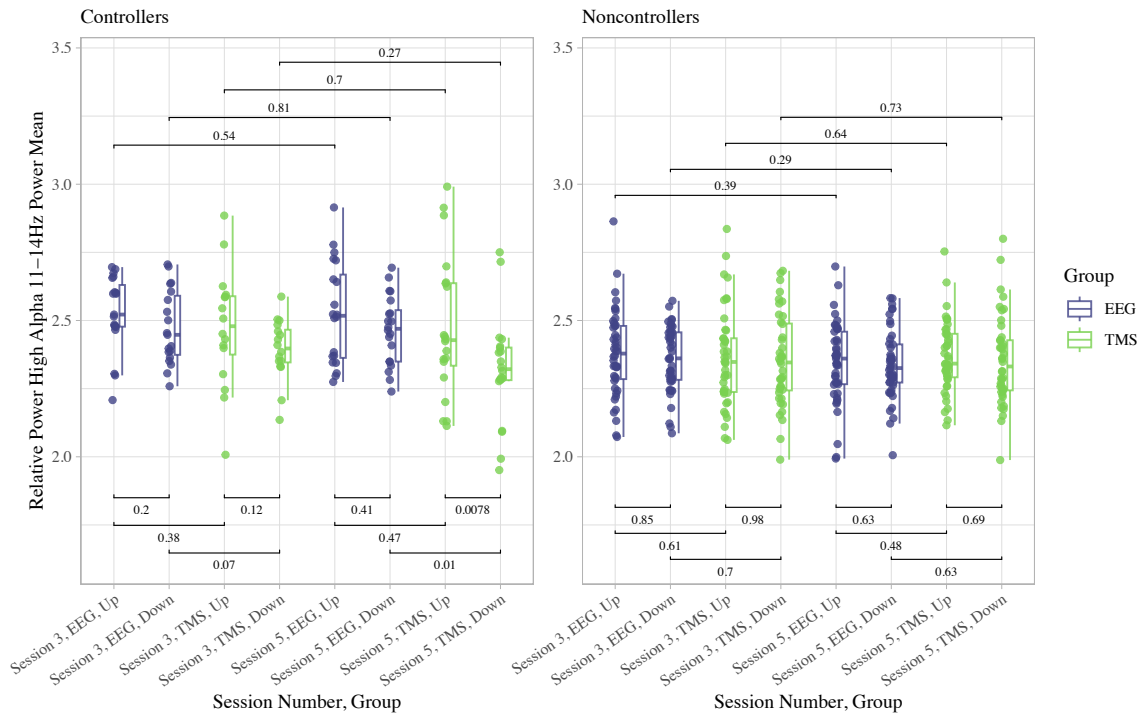


Figure 3.12. Relative Power Comparison TMS vs EEG Split. This figure shows the relative bandpower of high alpha on the y axis and session number, group, and condition on the x axis. The plot is split between BCI controllers (left) and noncontrollers. Additionally, the TMS group is coloured in green, and the EEG group in blue. Relevant statistical comparisons have been marked, no correction for multiple comparisons was performed. Visually a difference is visible in the third and fifth session of the TMS group and between the conditions and between the TMS and EEG groups. No comparison is significant in the Noncontroller plot.

Results Pattern 2

All results pattern of category 2 were significantly associated with the results of Up and Down but not with CA. This was the case for the percentage change powerbands of high delta, low alpha and high alpha. This was also the case for the relative bandpower power bands of high delta and low beta. For ease of comparison, the results for high alpha percentage change on CA, but also on Up and Down are presented in table 3.14, representatively for all Category 2 results. The same figures and tables are available for high delta, low alpha percentage change as well as high delta and low beta relative bandpower in the appendix.

Table 3.14*Model Predicting CA Using Low Alpha (11-14Hz) Percentage Change*

	CA			Up			Down		
	Beta	t-Value	p-Value	Beta	t-Value	p-Value	Beta	t-Value	p-Value
(Intercept)	61.74	20.46	0.001 >	9.77	9.44	0.001 >	8.76	8.14	0.001 >
Group (TMS)	-8.87	-2.04	0.044	-4.11	-2.69	0.0075	1.62	1.02	0.31
Time	-0.12	-0.91	0.36	-0.034	-0.57	0.57	-0.0027	-0.046	0.96
High Alpha	0.08	1.05	0.3	-0.076	-2.24	0.025	0.1	3.02	0.0026
Group (TMS):Time	0.6	3.01	0.0027	0.29	3.27	0.0011	-0.12	-1.35	0.18
Group (TMS):High Alpha	-0.1	-1.18	0.24	0.073	1.9	0.058	-0.11	-2.8	0.0052
Time:High Alpha	-0.0053	-0.99	0.32	0.0052	2.19	0.029	-0.0068	-2.96	0.0032
Group (TMS):Time:High Alpha	0.0069	1.16	0.25	-0.0049	-1.87	0.061	0.0071	2.74	0.0064

No high alpha main effect or interaction is significant for CA. However, for the Up MI both the main effect and the interaction with time is significant. Similarly for the Down MI the main effect and the interactions with group, time, and group and time is significant. It is interesting to note that in the Down MI, neither group nor training time is significant. However, the interaction with time and group, which is also a training time effect is significant. This may point towards high alpha power regulation being a relevant learning mechanism of this BCI that may be more important for the Down MI than the UP MI. The results for high alpha percentage change on CA in figure 3.13 seem to visually support this hypothesis.

Association Between Percentage Change, High Alpha 11–14 Hz Power And CA Split by Session, Control and Group

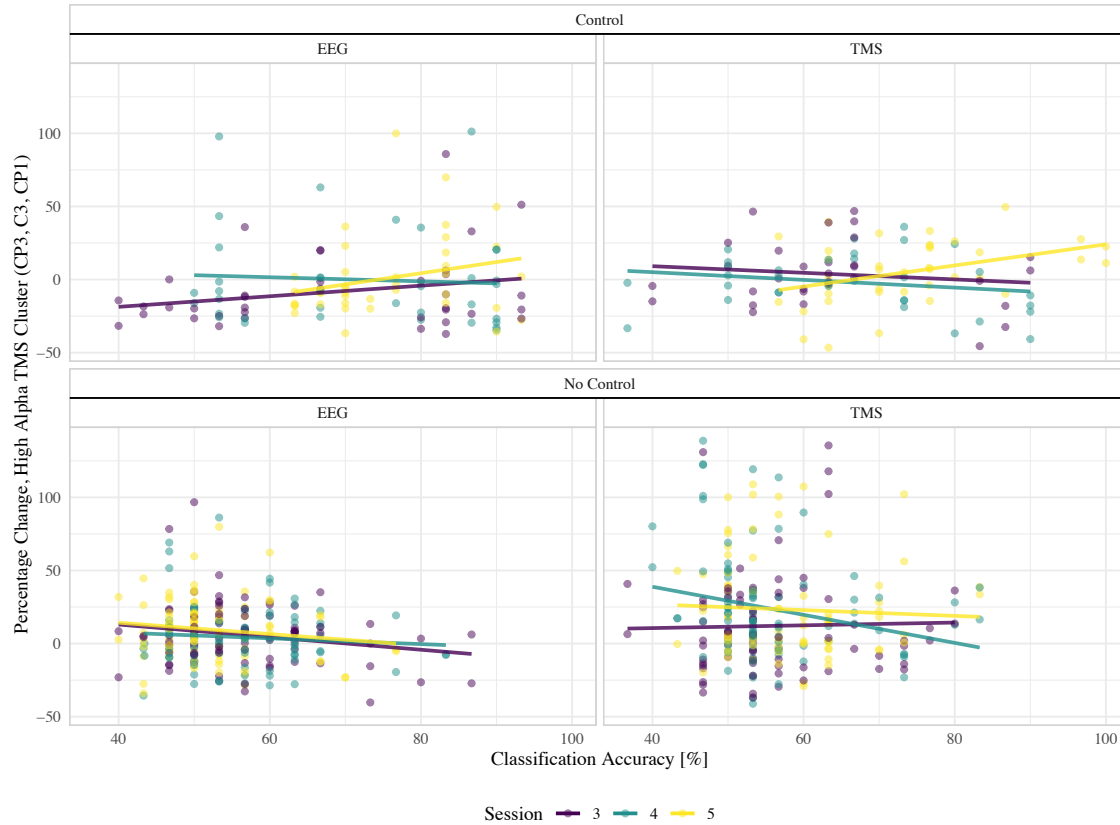


Figure 3.13. Percentage Change High Alpha Split Comparison Group and Control. This figure shows the low alpha mean power on the y axis and classification accuracy on the x axis. The plots are split by group with EEG in the left and TMS in the right column. The plots are also split by control achievers (top row) and noncontrollers (bottom row). Lastly, The different sessions are coloured. The controller plots suggest that there is a relationship between higher percentage change and higher CA, especially towards later sessions.

Plotting the power values in boxplots confirms a few other differences between groups and sessions (Figure 3.14). A comparison using a paired t-test of the TMS group at session five between the Up ($M = -2.88$, $SD = 15.5$) and Down ($M = 12.78$, $SD = 23.81$), $t(19) = -2.7$, $p = 0.014$. This difference is already present at session 3, (Up: $M = -3.7$, $SD = 20.6$; Down: $M = 11.13$, $SD = 20.81$) $t(15) = -2.4$, $p = 0.03$. The EEG group also has a significant difference between Up ($M = -12.58$, $SD = 14.2$) and Down ($M = 4.6$, $SD = 24.78$) at Session 5 $t(17) = -3.27$, $p = 0.0045$, but not at session 3.

Percentage Change, TMS vs EEG Group, Comparison of Third Session vs Last Session

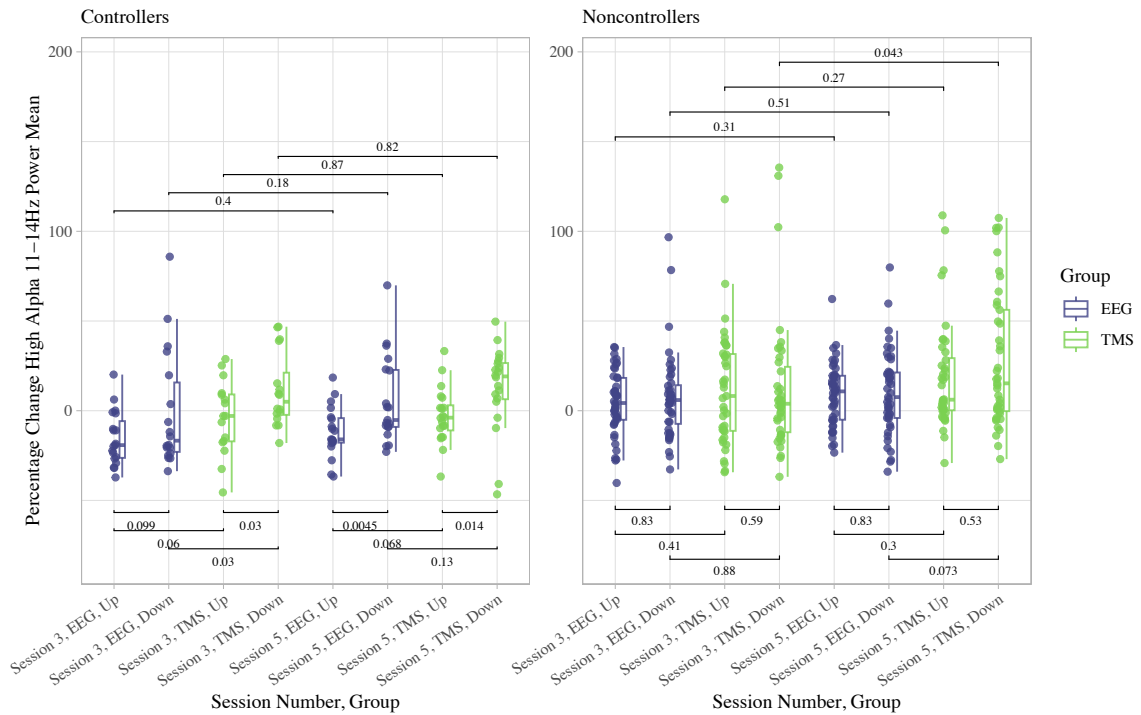


Figure 3.14. Percentage Change High Alpha Split Comparison Control. This figure shows the low alpha mean power on the y axis and session number, group, and condition on the x axis. The plot is split between BCI controllers (left) and noncontrollers. Additionally, the TMS group is coloured in green, and the EEG group in blue. Statistical comparisons have been marked. Visually and statistically a difference is visible in the last session, the TMS group and the Down condition. No other comparison is significant.

Results Pattern 3

All results patterns of category 3 were significantly associated with either results one of the MI states or CA. This was the case for the percentage change powerbands theta. This was also the case for the relative bandpower powerbands of high beta. The results for high beta RB on CA are presented in table 3.15. While theta had a significant term, as can be seen in the appendix, it did only have one significant term over all three models.

Table 3.15*Model Predicting CA Using High Beta (22-31Hz) Percentage Change*

	CA			Up			Down		
	Beta	t-Value	p-Value	Beta	t-Value	p-Value	Beta	t-Value	p-Value
(Intercept)	-97.10	-1.25	0.21	5.76	0.17	0.87	-34.64	-1.03	0.30
Group (TMS)	272.02	2.51	0.012	17.83	0.37	0.71	57.40	1.23	0.22
Time	7.92	1.51	0.13	0.67	0.29	0.77	1.72	0.76	0.45
High Beta	74.21	2.04	0.041	1.72	0.11	0.91	20.43	1.31	0.19
Group (TMS):Time	-16.33	-2.26	0.024	-2.51	-0.79	0.43	-1.92	-0.61	0.54
Group (TMS):High Beta	-131.09	-2.61	0.0094	-10.09	-0.45	0.65	-26.19	-1.20	0.23
Time:High Beta	-3.76	-1.53	0.13	-0.32	-0.29	0.77	-0.82	-0.78	0.44
Group (TMS):Time:High Beta	7.90	2.35	0.019	1.30	0.87	0.38	0.85	0.59	0.56

A high beta main effect is observable, so is an interaction with the group variable and an interaction between group and learning time. However, high beta relative is not significantly associated with either Up or Down MI, when using our linear models. This may indicate that the relationship between high beta percentage change and Up or Down MI is not linear. Another alternative suggested by figure 3.15 may be that high beta percentage change is relevant for a small group of participants within the model.



Figure 3.15. Percentage Change High Beta Split Comparison Control and Group. This figure shows the high beta power on the y axis and classification accuracy on the x axis. The plots are split by group with EEG in the left and TMS in the right column. The plots are also split by control achievers (top row) and noncontrollers (bottom row). Lastly, the different sessions are coloured. The controller plots suggest that there is a relationship between higher percentage change and higher CA, especially towards later sessions for the TMS controller group.

A split by conditions does not lead to any significant differences for percentage change high beta.

Results Pattern 4

The only result of this category was low beta percentage change for the results of Down MI.

Table 3.16*Model Predicting CA Using High Beta (22-31Hz) Percentage Change*

	CA			Up			Down		
	Beta	t-Value	p-Value	Beta	t-Value	p-Value	Beta	t-Value	p-Value
(Intercept)	62.58	20.55	0.001 >	9.15	8.7	0.001 >	9.59	8.78	0.001 >
Group (TMS)	-9.73	-2.25	0.027	-3.49	-2.32	0.021	0.78	0.5	0.62
Time	-0.17	-1.28	0.2	0.0055	0.091	0.93	-0.056	-0.95	0.34
Low Beta	0.19	1.26	0.21	-0.11	-1.64	0.1	0.16	2.51	0.012
Group (TMS):Time	0.65	3.35	0.001 >	0.25	2.94	0.0034	-0.069	-0.81	0.42
Group (TMS):Low Beta	-0.29	-1.76	0.078	0.078	1.07	0.28	-0.17	-2.32	0.021
Time:Low Beta	-0.013	-1.34	0.18	0.0072	1.75	0.08	-0.011	-2.75	0.0062
Group (TMS):Time:Low Beta	0.019	1.82	0.069	-0.0058	-1.24	0.22	0.012	2.53	0.012

No low beta main effect or interaction is significant for CA or Up. However, for the Down MI both the main effect and all the interactions with time are significant. It is interesting to note that in the Down MI, neither group nor training time is significant. However, the interaction with time and group, which is also a training time effect is significant. This may point towards low beta power regulation being a relevant learning mechanism of this BCI that may be more important for the Down MI than the UP MI. The results for high alpha percentage change on CA in figure 3.13 seem visually to support this hypothesis. This result is quite similar to result category 2 for high alpha. The similar results may be linked to the similarity of the frequencies.

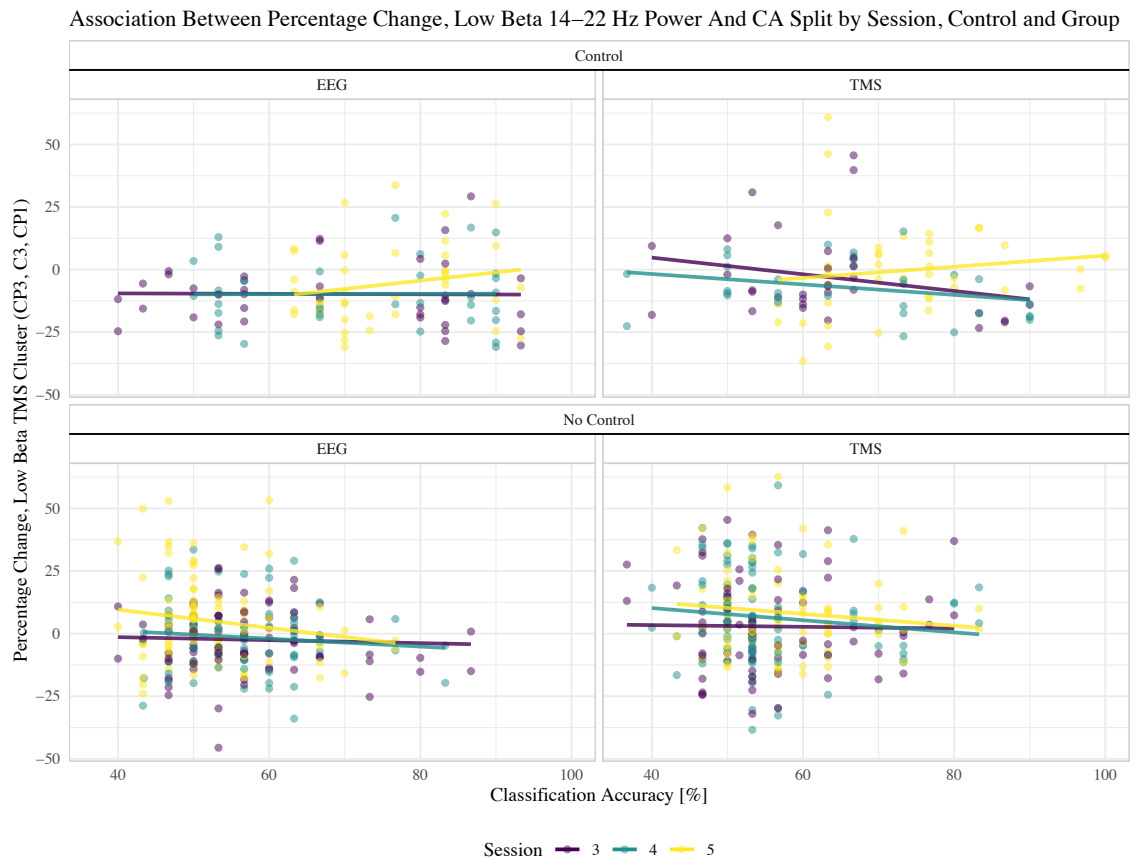


Figure 3.16. Percentage Change Low Beta Split Comparison Control Group. This figure shows the low beta power on the y axis and classification accuracy on the x axis. The plots are split by group with EEG in the left and TMS in the right column. The plots are also split by control achievers (top row) and noncontrollers (bottom row). Lastly, The different sessions are coloured. The controller plots suggest that there is a relationship between higher percentage change and higher CA, especially towards later sessions for the TMS controller group.

Figure 3.16 shows a pattern for EEG and TMS. When we further split up the data into Up and Down conditions (Figure 3.17) we can see that EEG controllers have clear differences in relative bandpower high alpha (Up: $M = -11.3$, $SD = 14.08$; Down: $M = -0.62$, $SD = 16.18$, $t(19) = -2.38$, $p = 0.028$, using a paired t-test. Further, there is a difference between the EEG Down ($M = -14.43$, $SD = 10.71$) and Up ($M = -5.083$, $SD = 13.2$) condition $t(19) = -2.62$, $p = 0.017$. This difference may be present in the TMS group, but if so it is less pronounced. The noncontrollers have no significant comparisons.

Percentage Change, TMS vs EEG Group, Comparison of Third Session vs Last Session

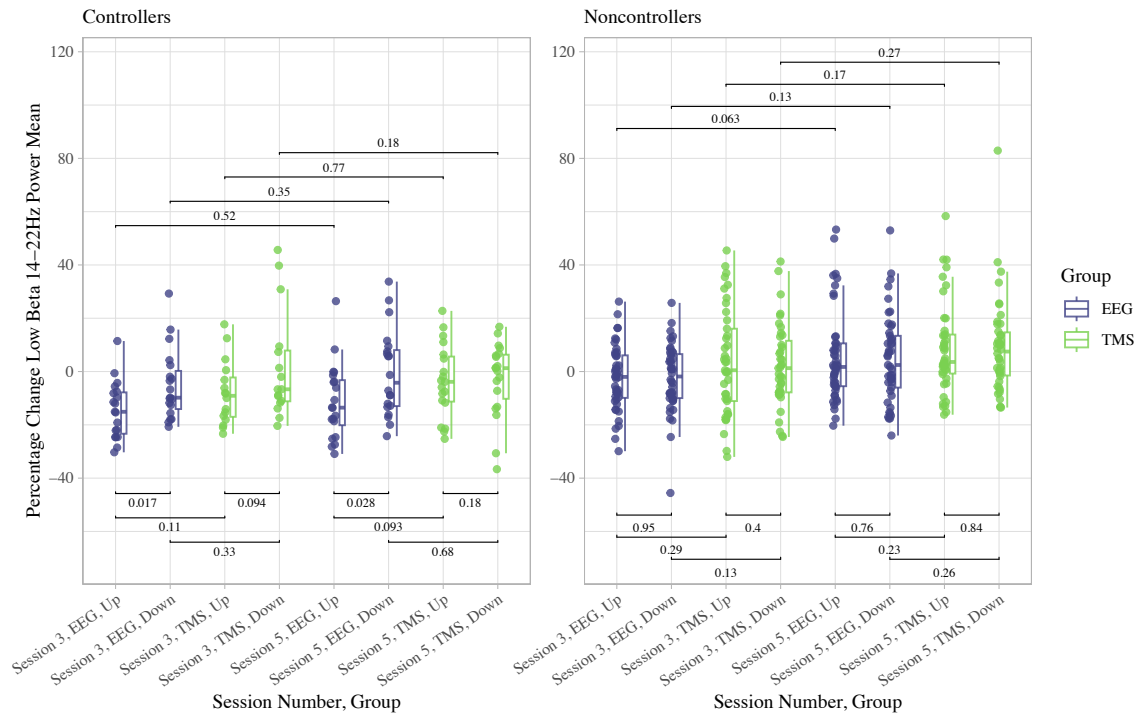


Figure 3.17. Percentage Change Low Beta Split Control. This figure shows the percentage change of low beta on the y axis and session number, group, and condition on the x axis. The plot is split between BCI controllers (left) and noncontrollers. Additionally, the TMS group is coloured in green, and the EEG group in blue. Statistical comparisons have been marked. Visually and statistically a difference is visible in the last session, the TMS group and the Down condition. No other comparison is significant.

Results Pattern 5

Results pattern 5 showed no significant association to either CA, or the MI states. This was the case for percentage change high beta, and low gamma and for relative bandpower theta (see appendix). The fact that both percentage change high beta and low gamma have no significant associations, but relative bandpower low gamma and high beta do may signify that different aspects of the power spectrum may be relevant for BCI control.

3.2.5 Discussion

Main Findings

In this study we demonstrated that participants in the Transcranial Magnetic Stimulation (TMS) group were able to learn how to use the Brain-Computer Interface (BCI) faster, and had better scores than the Electroencephalography (EEG) group after five sessions, demonstrating that TMS-Neurofeedback (TMS-NF) may be a useful adjunct to prime users for EEG BCI. Further, participants were able to regulate their Motor Evoked Potential (MEP) amplitudes using two different Motor Imagery (MI) strategies involving the same limb under more difficult conditions than previously shown (Ruddy et al., 2018). The data also suggest that performance of Up MI profited more from the TMS-NF priming than the Down MI, which is promising as the ultimate goal for stroke rehabilitation would preferentially target up-regulation of corticospinal pathways to prime the motor pathways for neuroplasticity.

The group primed with TMS-NF prior to EEG BCI had a different scalp distribution of neural activity during BCI performance, with Common Spatial Pattern (CSP) weights distribution showing a left lateralised pattern for the Down condition, and a central band from left to right for the Up condition. In comparison, the unprimed group showed a frontal and right lateralised weight distribution for the Down condition, and a narrower band oriented in posterior to anterior. We found that distinctiveness between the neural activity associated with the two different MI strategies was associated with higher Classification Accuracy (CA), but this was not affected by TMS-NF priming. Aside from spatial patterns, we found that stability of neural patterns was neither different between MI conditions nor between the groups.

Probing deeper into the EEG signatures associated with the MI conditions and BCI performance quality, results indicated that signals within the alpha range were more affected during Up MI performance, and beta was more related to Down. This is consistent with the prevailing view that beta has a key role in motor inhibition (Enz et al., 2021; Little et al., 2019; Wagner et al., 2018). We also found that low gamma frequencies were related to MI success.

Additional Findings

We found that while background Electromyogram (EMG) activity (BrMS) was significantly associated with MEP amplitude, the training time effect was a statistically independent effect. Further, using dynamically thresholded MEPs reduced the impact of BrMS on MEP amplitude. We could show that MI ability increased MEP amplitude,

implying that MI ability may be linked to poststroke recovery (Stinear et al., 2006a; Stinear et al., 2017b, 2020).

We confirmed most of our preregistered hypotheses, showing that participants primed using TMS-NF were able to learn to control the EEG BCI faster and with better performance (H2, H2.4). We additionally demonstrated that they started with a better performance in their first EEG BCI session, suggesting that the TMS-NF priming gave them an initial advantage (H2.1) and that this advantage was sustained, resulting in better performance also in their fifth and final session compared to the control group who undertook the same amount of training but only with the EEG BCI (H2.3). However, when comparing both groups at their third total session (meaning: the first EEG BCI session for the TMS group vs the third EEG BCI session for the EEG group) there was no significant difference (H2.2). Importantly for future BCI design and implementation, we found that sleepiness (Karolinska Sleepiness Scale, or KSS score) was associated with BCI performance, suggesting that mental fatigue should be closely monitored in any BCI experiment. The negative effect of the KSS was also true for stability and distinctiveness, which may provide insight into the mechanism by which fatigue affects BCI performance.

While we could successfully confirm our hypotheses, overall performance on the BCI as measured with CA was low. Of 34 participants only 10 could be classified as controllers, each group having 5 controllers. This is lower than control-based BCI literature, describing around 10%-30% of BCI inefficient participants (Allison & Neuper, 2010; Vidaurre & Blankertz, 2010; Lotte & Jeunet, 2015). It is also lower than a similar study we conducted, testing the feasibility of this setup with a wearable 16 channel, dry electrode, Bluetooth EEG headset, which had 50% controllers (Simon & Ruddy, 2022).

There are several issues that could explain these results. It could be that we offered too few training sessions with this BCI. It could be due to overfitting, as our CSP produced 16 features given our relatively small training sample size of 15 trials per condition (30 total). There were also slight differences between the calibration and training stages of the BCI. During the calibration no feedback – in form of a blue bar going up or down – was available. This may have introduced noise in the EEG signal by producing signal in visual areas not stimulated during the calibration, and it may have introduced a source of distraction for the user. Lastly, in this study we showed that mental fatigue (as measured with the KSS Score) heavily influenced CA scores. The more complex EEG setup, as well as the fixed times participants were required to attend sessions may have led to worse performance than when participants have more autonomy in choosing when to train, and can setup the equipment

faster, as was the case in the 2022 study by Simon and Ruddy. These aspects should be improved in future BCI designs, as they may influence the EEG data on which further analysis is based. We analysed the EEG spectral data using the two different metrics, the percentage change from baseline and relative bandpower. We further divided our available data into seven power bands: High delta (1-5Hz), theta (5-8Hz), low alpha (8-11Hz), high alpha (11-14Hz), low beta (14-22Hz), high beta (22-31Hz), and low gamma (31-45Hz). Additionally, since CA was composed of two different MI states (Up and Down) and could be affected independently we computed 42 (2 metrics * 7 Powerbands * 3 outcomes) models.

Of these 42 results, 23 had a power band component that was significantly associated to an outcome measure of CA, Up or Down successes. We further categorised these 42 results into five different patterns from category 1: where powerband components were significantly related to all outcomes, and category 5 of no powerband component being significant. Percentage change and relative bandpower showed different patterns across the categories, further emphasising the importance of processing EEG data.

Covariates of MEP

We could show that the effect of training was independent of the BrMS despite the connection between the BrMS and the average MEP amplitude being well documented in literature (Devanne et al., 1997; Hasegawa et al., 2001; Hess et al., 1986). Because of this potential confounding variable we addressed it on three different levels. During the experiment, participants were not allowed to proceed to the stimulation phase of the TMS-NF until their BrMS reached a level below 3.5 microvolt. During the preprocessing of the MEP data any MEP with BrMS values above ten microvolt were excluded from analysis. Lastly, we used linear models which are powerful statistical models able to adapt to a wide variety of influences and provide robust outcomes (Boisgontier & Cheval, 2016; Gueorguieva & Krystal, 2004; Wainwright et al., 2007; Yang et al., 2014; Yu et al., 2022).

Due to the link between the biological systems of BrMS and MEPs as well as it being measured on the same device, it is unlikely that these two effects can be fully dissociated. However, with strong statistical tests and high quality data we could show that while BrMS was linked to MEP amplitude directly, it was not linked to the effects of the condition and training variables. Further, we showed that more complex models and variables taking into account the variability of MEP amplitudes could significantly reduce the effect of BrMS on the MEP amplitude.

Lastly, while our data may not be able to settle an ongoing debate on MEP variability, our data strongly suggests that a participant's brain state during MI is relevant. Recent literature still suggests that power, phase, and different processing methods for the online assessment of power and phase *may or may not* be responsible for MEP variability (Bigoni et al., 2023; Karabanov et al., 2021; Keute et al., 2023; Madsen et al., 2019; Poorganji et al., 2023; Wischniewski et al., 2022). Our data, and data from similar studies (Mihelj et al., 2021; Ruddy et al., 2018), shows that MI and therefore the ongoing mental processes in a participant during the TMS play a decisive part in MEP variability.

CA and EEG Measures

The difference between percentage change and relative bandpower is easily explained when thought in terms of what is happening during the BCI task. The EEG BCI categorises brain activity using a Common Spatial Filter (CSP) and a Linear Discriminant Analysis (LDA). The input into this CSP and LDA is the squared logarithmic power of the EEG signal filtered from 8-30Hz, which is similar to the procedure for relative bandpower. Further, the LDA is a process trying to establish the difference (maximised by the CSP) between two states which is similar to what a linear model is attempting.

It is not surprising that we find significant results for relative bandpower, low alpha and high alpha, as it is well described that MI instructions about hand movement typically result in activity centred around the mu band (Neuper et al., 2009; Pfurtscheller et al., 2003, 2006). However, it is surprising that low and high beta have different patterns of significance. Beta power has been described to desynchronise during MI and rebound (resynchronise) after MI has been executed (Pfurtscheller & Lopes da Silva, 1999). It may be that our relatively long period of analysis (3s) is too long to reliably pick up short bursts of beta synchronisation, or that the synchronisation and desynchronisation balance each other out. Further, the long held view is that beta power desynchronises during and rebounds at the end of MI (Pfurtscheller et al., 1996), and as we instructed participants to continuously perform MI until the fixation cross disappeared this effect may have weakened or led to averaged beta rebounds during our period of analysis.

Similarly to alpha and beta, low gamma has been associated with BCI control in literature and its association to BCI outcomes is therefore supported. Low gamma has been correlated with good BCI performance in other experiments (Ahn et al., 2013a). Low gamma increases have been described with increasing MI task complexity (Mashat et al., 2019). Further, gamma, theta and low alpha have been implicated in BCI performance (Ahn et al., 2013b). Low gamma was only significant for the metric of relative bandpower. This could

be due to qualities of its calculation, like logarithmic transformation in relative bandpower processing, which is more sensitive to large changes than the percentage based transformation, or it could be due to the nonnegative quality of relative bandpower.

Given the significant association of the KSS on almost all outcomes, we suggest that some of the results we report, are likely proxy or composite measures of attention and sleepiness. Especially as sleep onset is associated with a variety of fast paced changes in low frequency (delta, theta and alpha) changes (Fernandez Guerrero & Achermann, 2019).

Delta oscillations synchronisation at the start and end of trials is linked to MI (Barrios et al., 2019). The frequency range has other rather unique qualities such as high signal-to-noise ratio and is able to predict gender, which are being investigated (Abenna et al., 2023; Kaur et al., 2019), but there is little on the contribution of delta frequencies to BCI control. This made it difficult to compare it with our results that show delta as significantly associated to the Up and Down MI in both percentage change and relative bandpower. However, Ruddy et al. (2018), who used the same MI instructions and TMS-NF, showed that delta was significantly different between Up and Down MI.

The role of theta in this experiment is likely linked to attention rather than to the MI execution. Theta seems to play a role in cognitive processing (Harmony, 2013; Ishii et al., 2014) and is linked to spatial attention (Fiebelkorn & Kastner, 2019) and MI performance (Trambaiolli et al., 2019). Further, spontaneous theta fluctuations have been observed at sleep onset (Kotajima et al., 2005). Therefore, theta seems to be linked with attention and the lack thereof, explaining its influence on BCI performance. This could explain why as a percentage change metric it is significantly associated to CA but not with either Up or Down MI and why the relative bandpower which is closer to the BCI classification mechanism, does not reach significance.

Overall with all measures in category 2 (percentage change: high delta, low alpha and high alpha; relative bandpower: high delta, and low beta) it is possible that the compound nature of the outcome measure is problematic. CA is a measure of successful Up trials and successful Down trials, therefore if for example high delta had a positive effect on Up trials, but an equally negative effect on Down trials the overall averaged effect on CA may be lost. This effect may also explain why there are more significant results in relative bandpower as it is a nonnegative measure.

As the selection of electrodes for the powerband analysis was based on a preliminary analysis of the CSP weights distribution, this analysis should be examined critically. The

improved and reported CSP analysis does show that the selected cluster (C3, CP3, CP1) may not have captured the highest difference between the conditions or groups.

Improved Outcome for the TMS Group

We found that the CSPs of the TMS group controllers was visually different from the CSPs of the EEG group controllers. While both groups had higher weight distribution centrally for the Up condition, the TMS group had a left to right central band, while the EEG group had a narrow posterior to anterior band. In the Down condition the TMS group had a left lateralised pattern, while the EEG group had a frontal and right lateralised pattern of CSP weight distribution.

While EEG data is not optimal for drawing topological conclusions, we think that this result is noteworthy. This is because during the up condition the EEG group distribution is likely to correspond to the primary motor and premotor cortex, while the TMS group distribution overlaps more with the primary motor and the sensory motor cortex. This interpretation is supported by current literature.

Porro et al. (2000) studied differences in cortical activation between MI and Motor Execution (ME) using functional Magnetic Resonance Imaging (fMRI). They found that activations of the motor and somatosensory cortex were stronger in ME than MI and that activation of the supplementary motor area and premotor cortex were stronger in MI than in ME. Others have found similar results (Decety, 1996; Dechent et al., 2004; Héту et al., 2013).

However, the motor cortex may be more active in MI under certain conditions (Lebon et al., 2012) such as after a stroke (Confalonieri et al., 2012; Di Rienzo et al., 2014). The suggestion that TMS-NF training may be able to mimic activations seen in ME may have big implications for potential rehabilitative applications. The rehabilitative prospects of TMS-NF are further enhanced through somatosensory input during MI, which may increase learning-induced plasticity (Bonassi et al., 2017).

Strengths and Limitations

Our analysis was preregistered, and we used the current scientific gold standard statistical tools and open science practises. This should increase the robustness, validity and reproducibility of our results. Compared to other BCI investigations, this study had a relatively large sample size and benefits also from the longitudinal nature of the training data. The data includes 34 participants for five sessions each averaging 10-15 hours of training. Further, we collected control variables such as MI ability which we used to balance the groups and measured levels of fatigue during BCI training. We also implemented three

different ways (behavioural, preprocessing and statistical) to control for the influence of muscle activity during TMS-NF.

Further, our results are consistent with those described in other literature with similar BCI setups, increasing the validity of our results. Our data suggest that priming using TMS-NF may allow future applications to teach MI with very specific goals such as finger individuation (Mihelj et al., 2021). This should allow for more precise targeting of the brain's motor rehabilitation mechanisms.

However, there are some points that warrant further investigation. The quantity of BCI 'controllers' was low, suggesting that the experiment setup may need optimisation. It may be that the clinically inspired MI instructions are difficult to execute. However, it may be that our choice of alpha and beta power bands combined with CSP and LDA may not be ideal for the accurate classification of this kind of MI task. Finally, our plan for experimenter blinding had to be partially abandoned to minimise personal contact during the COVID-19 pandemic.

Future studies should aim to increase BCI performance as measured with CA. This could be achieved by addressing the issues we described such as reducing the number of CSP features. Alternatively, the classifier calibration dataset could also be increased by collecting more data or by using alternative calibration procedures such as transfer learning (Arvaneh et al., 2017; Azab et al., 2018, 2019). The classifier could also be allowed to recalibrate whenever new data is available. Protocols could be altered to allow for more visual consistency between the calibration and training trials. This could be tied to other BCI improvements centred on the human experience of the BCI, such as delivering more meaningful feedback, and giving users more control over their training (Jeunet et al., 2020).

Outlook

Improved care and rehabilitation for stroke survivors may be critical to combat rising healthcare costs due to changing demographics. The efficiency of this BCI should be investigated in a clinical population, especially as it enables training during the critical phase of early poststroke recovery, possibly synergising with biological recovery processes.

However, the advantages of multimodal and multiphasic BCI may not be limited to rehabilitative applications. There are many other applications of MI-based BCI that, could profit from this approach to improve classification accuracies.

Further, the success rate for these MI instructions should be improved. There are several ways this optimisation can be achieved. The BCI classification process, using CSP

and LDA, could make use of transfer learning to speed up calibration times (Arvaneh et al., 2017; Azab et al., 2018, 2019). Additionally, the BCI could be taught to recognise errors or inattention from the user to increase the accuracy of individual trials (Martel et al., 2016; Wirth et al., 2019). Another approach would be to focus on the human using the BCI. By optimising instructions, feedback and the usability of the BCI rates of success could be increased (Jeunet et al., 2020; Roc et al., 2021).

Conclusion

The results presented here suggest that TMS-NF may be a valuable adjunct to prime users prior to learning an EEG BCI to improve the subsequent rate at which BCI control is learned, and ultimately result in better BCI performance. This may further contribute to reduced rates of BCI illiteracy, thus maximising the proportion of users that can gain benefit. Applied to upper limb rehabilitation following stroke, the results suggest that TMS-NF could be used for two sessions at the patient's bedside to provide training on how to best harness motor imagery to upregulate excitability in surviving corticospinal pathways, to maintain neural traffic in motor circuitry and promote neuroplasticity. We envisage a future 2-phase approach where the patient is additionally issued with a wireless, wearable EEG BCI that operates upon driving signals derived from the same mental imagery strategies practised with the TMS-NF. This anticipates to facilitate long term at-home rehabilitation to promote the stroke survivor's independence.

3.2.6 Author Contributions

Colin Simon: Conceptualization, Methodology, Software, Data collection, Formal analysis, Investigation, Writing, Visualization, Project administration. **Olga Akhmetova:** Data Collection. **Vivienne Andrea Simon:** Project Administration, Writing - review & editing. **Kathy Ruddy:** Conceptualization, Methodology, Software, Writing - review & editing.

CRedit Contributions

Conceptualization: C.S.; Data curation: C.S.; Formal analysis: C.S.; Funding acquisition: K.R.; Investigation: C.S. and O.A.; Methodology: C.S. and K.R.; Project administration: C.S., K.R., V.A.S. and O.A.; Resources: C.S., K.R., V.A.S. and O.A.; Software: C.S. and K.R.; Supervision: C.S. and K.R.; Visualization: C.S.; Writing – original draft: C.S.; Writing - review & editing: C.S., K.R. and V.A.S.

4. The Feasibility of Multiphase BCI Rehabilitation

4.1 Chapter Introduction

The previous chapter explored opportunities to increase the success of poststroke motor rehabilitation using a combination of two BCI modalities. The increased complexity of this approach means that the resulting system may be best used early after stroke when trained administrators are more readily available (i.e. at the bedside).

A streamlined version of the EEG BCI suggested in chapter three may be simple enough for stroke survivors to take home and train without supervision. In this chapter we tested the feasibility of such a simplified EEG BCI by sending EEG hardware and software home to healthy participants with instructions on how to execute at-home BCI training.

4.2 A Wireless, Wearable Brain-Computer Interface for Neurorehabilitation at Home; A Feasibility Study

4.2.1 Abstract

Despite Brain-Computer Interfaces (BCI) showing potential for motor rehabilitation following stroke, clinical adoption is lacking. BCI featuring unsupervised training at home may facilitate the clinical use of BCI. This study tested the feasibility of a wearable, 16 dry electrode wireless electroencephalography (EEG) system for BCI training with healthy participants. Twelve participants trained for six days using an EEG BCI at home, receiving remote instructions. A Graz style two-state motor imagery training protocol was used, requiring two different mental strategies. The BCI used common spatial patterns and linear discriminant analysis to classify mental states and provide feedback. The primary outcome was online classification accuracy of the mental state during BCI training, and secondary outcomes were the stability and distinctiveness of the EEG signal during mental states as well as the changes in alpha (8-13Hz) and beta (14-30Hz) power due to training. Six participants achieved control over the BCI. The distinctiveness of the EEG signal between the states increased over time and was associated with success. Stability increased over time but only in one mental state. Successful control was characterized by activity changes in alpha and beta power. Therefore, participants could achieve voluntary control over a BCI within six days using a wireless, wearable dry EEG system at home with minimal expert instruction or supervision.

4.2.2 Introduction

Stroke is a leading contributor to adult disability (Virani et al., 2020). With rising population age (United Nations et al., 2017) an increase of stroke and therefore disability is expected (Lawrence et al., 2001; Nakayama et al., 1994; Virani et al., 2020). While stroke mortality has decreased, the amount of survivors with disabilities has increased as a consequence (Vos et al., 2020).

Brain-Computer Interfaces (BCI) show potential for easing the healthcare burden of stroke. Small scale clinical trials using BCIs for stroke rehabilitation have shown promising results, yet large-scale clinical adoption is lacking (Simon et al., 2021). A possible remedy for the slow uptake of BCI are multi-phase BCIs, featuring an unsupervised training phase at home, which reduces the need for expert supervision and the duration of the hospital stay, and increases the amount of training and agency for the patient (Simon et al., 2021). To promote this novel approach, it is necessary to establish the feasibility of wearable, convenient electroencephalography (EEG) systems for BCI. This study tested the feasibility of a convenient, 16 dry electrode wireless EEG system for a BCI intervention at home with healthy participants.

Adaptations to a mainstream BCI scenario in OpenViBE (Renard et al., 2010) were made for a BCI protocol that would be suitable for stroke rehabilitation using mental imagery strategies previously found to be effective for altering the responsiveness of the corticospinal pathways (Liang et al., 2020).

The primary hypothesis of this study was that participants would achieve control over the BCI within six training sessions. The secondary hypotheses were that participants would be able to regulate their brain activity measured by the power of alpha (8-13Hz) and beta (14-30Hz) frequency bands as well as increase the stability and distinctiveness of their EEG signals in the two different mental states.

4.2.3 Methods

Participants

A total of 13 participants with no pre-existing training were recruited for this feasibility study. One participant was discontinued after 3 sessions due to continuously bad EEG signal quality, especially in the occipital region likely due to the wrong EEG cap size and thick hair. All participants were right-handed. Ethical approval was granted by the

School of Psychology Ethics Committee of Trinity College Dublin and conformed to the rules laid out in the declaration of Helsinki.

Setup and Equipment

Data collection was performed by participants in their homes under remote instruction and supervision from the research team using Skype, Zoom and Microsoft Teams.

Participants were instructed on how to measure their head for the appropriate EEG cap size via email. After participants had received the EEG Cap, an online video call was scheduled. Before the call, participants received access to an online folder containing the software used for the BCI, OpenViBE (Renard et al., 2010), as well as instruction documents on how to install OpenViBE, and how to run the experiment. Participants were instructed to try to install the software, as well as run the experiment before the video-call. During the first video-call the administrators made sure the participants had set up the equipment correctly and talked them through the experiment.

After the first video call, participants were asked to run the experiment once, upload their data and reschedule for a second video-call. Before the second video-call, administrators inspected the data collected by participants. During the second video-call the administrator asked to be talked through the data collection procedure the participant had implemented alone to make sure the data was collected correctly. If no issues were identified participants were encouraged to continue the data collection. Participants were encouraged to contact the research team should there be any question regarding the data collection or experimental procedure. Lastly, participants were encouraged to continuously upload their data to a specified folder, so it could be checked by experimenters. Participants were only contacted again by administrators upon the conclusion of the data collection.

Participants were instructed to sit in a comfortable chair and to remain relaxed throughout the experiment, and to move as little as possible. A 16 channel EEG cap (Iz, O2, Pz, P3, CPz, P4, Cz, FC3, FCz, FC4, Fz, FT7, AF7, FPz, AF8, FT8) with dry contact electrodes (Cumulus, Belfast, UK), was used to record data and transmit signals in real-time to a computer via Bluetooth. Participants were instructed to seek help from another adult to measure their head and find the vertex: From nasion toinion, and from left to right periaucular points. The EEG cap was placed so that the central electrode (Cz) was over the vertex. Reference electrodes were placed over the mastoids using adhesive disposable electrodes.

BCI Overview

The employed EEG BCI protocol was a custom modification of the well-established Graz-BCI (Pfurtscheller et al., 2003) as implemented in OpenViBE. The Graz-BCI aims to detect differences in real-time between a participant's performance on two mental imagery tasks: moving the left and moving the right hand. To test the feasibility of a BCI for neurorehabilitation at home, the traditional instructions were adapted to be more clinically relevant, with instructions shown to change corticospinal excitability in stroke-affected limbs (Liang et al., 2020). This resulted in two mental imagery instructions: Up and Down. For the Up state, participants were asked to imagine complex or forceful movements of fingers on the right hand, which has been shown to upregulate corticospinal excitability of the pathways connecting the motor cortex to the muscles (Ruddy et al., 2018). For the Down state, participants were asked to imagine that the right hand and fingers were cold, dead or detached, which has been shown to downregulate excitability in the same pathways (Ruddy et al., 2018). Further, the onscreen feedback was adapted to suit this protocol; The original horizontal left-right feedback bar display was changed to a vertical up-down one, and to include a red hand symbol cue during Up trials and red circle cue with an X inside it for Down trials (Figure 4.1). The sessions were completed, if possible, on successive days resulting in a mean interval of 1.56 days. A BCI session was split into 3 phases: The first phase involved acquiring training data, the second phase consisted of training a Common Spatial Pattern (CSP) and a Linear Discriminant Analysis (LDA) with the acquired data to enable trials with feedback, and the third phase displayed online real-time feedback of ongoing neural activity to the participant, the format of which resembled the training phase but with a blue bar fluctuating vertically above or below a horizontal midline representing the LDA's confidence in the classification of the current brain state.

BCI Training and Feedback Format

Each of the six sessions commenced with one calibration block followed by four online feedback blocks. The calibration data was gathered in one block of 40 trials; 20 of each condition administered in a random sequence. For the Up trials the cue was a right hand, set above the horizontal midline. For Down trials the cue was a crossed circle below the midline (Figure 4.1).

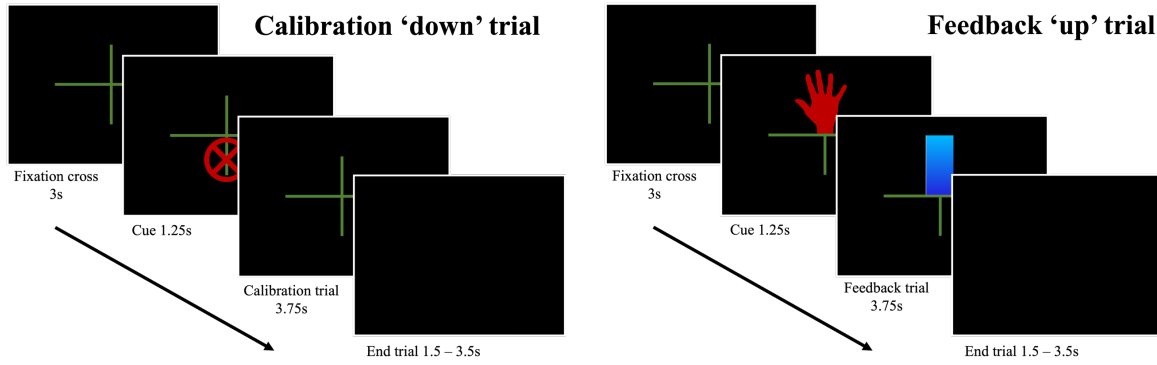


Figure 4.1. EEG BCI Protocol. Timeline of an EEG BCI trial for the calibration block (left) and online feedback blocks (right). Every session started with a calibration block without feedback to train the BCI and then proceeded with feedback trials. One complete trial was 9.5 - 11.5 seconds, and 40 trials made one block.

Each trial had four stages as shown in figure 4.1. First a green fixation cross appeared for three seconds followed by a cue for 1.25 seconds. The participant was instructed to commence the relevant mental imagery after the cue disappeared. In the initial training block, a green fixation cross was then shown for 3.75 seconds. During the online feedback trials a blue bar was shown fluctuating vertically up- or downwards from the midline, with the length of the bar representing confidence in classifying this trial. In the final stage a black screen appeared for a variable amount of time ranging between 1.5 to 3.5 seconds. A complete trial was between 9.5 to 11.5 seconds.

BCI Online Processing

The signal was acquired from the cap in chunks and bandpass filtered between 8-30Hz (Butterworth, 5th order, bandpass ripple of 1Hz). The signal was then transformed using the equations of the CSP, extracting eight features, acquired during the calibration stage. The signal was epoched into segments of 1s every 16th (0.0625) of a second. Within the epoch the logarithmic band power was calculated by squaring, averaging, and taking the logarithm. The LDA classified the signal according to the parameters trained with the calibration data. The classification confidence of the LDA was displayed to the user as feedback.

EEG Offline Preprocessing

To reduce bias, an automated process (Artifact Subspace Reconstruction, ASR) as implemented in EEGLAB (Version: 2021.0) (Delorme & Makeig, 2004) running on MATLAB (R2021a, Mathworks Inc) was used to process the raw EEG data.

ASR is a fully automatic state-of-the-art preprocessing method (Gabard-Durnam et al., 2018; Mullen et al., 2013, 2015). ASR identifies a sequence of relatively quiet (low standard deviation) EEG calibration data (Mullen et al., 2013, 2015). It then performs a 0.5s sliding window principal component analysis (PCA) of the whole signal and rejects sequences of signal whose standard deviation surpasses 20 times the standard deviation of the calibration data (Gabard-Durnam et al., 2018; Miyakoshi, 2021).

One block (Participant 7, session 5, block 2) was excluded from the analysis due to technical difficulties during recording.

The preprocessing encompassed the following steps: A lowpass filter centered around 45Hz and a highpass filter of 1Hz were applied to the signal and it was referenced to average to provide the cleanest possible input signal to ASR. The built-in automatic channel selection of the ASR rejected channels with more than ten seconds of silence, more than five standard deviations of line noise, and less than 0.6 correlation to the signal of neighboring channels. The ASR ran with default settings that are considered conservative (Chang et al., 2018; Gabard-Durnam et al., 2018). The default settings were set to: use Euclidian distance for calculations (rather than Riemannian), reject bursts (meaning that data was rejected rather than reconstructed), and a burst criterion of 20 (multiplication factor of the standard deviation of the calibrating data). Additional data removal criteria were set to default: data were removed if more than 25% of channels showed more than seven standard deviations of variance within a time window. The rejected channels were then spherically interpolated and added back to the signal that was referenced to average again. The ‘PICARD’ independent component analysis (ICA) ran with default parameters (Ablin et al., 2017). ICA components with 90% confidence of artefactual origin (Muscle, Heart, Line Noise or Channel Noise) were rejected automatically. A trained experimenter visually inspected the raw data and processed outputs to ensure the automatic processing was performing as expected and detailed performance measures were recorded. The signal was cut into the 2 spectral bands OpenViBE used for classification: the alpha band (8-13Hz) and the beta band (14-30Hz). Those bands were analysed as measured at the FC3 electrode, chosen as it was closest to the left motor cortex, and as measured between the FC3 – FC4 electrode chosen as an operationalization of lateralization.

The data were compared using independent sample t-tests between groups (or Wilcoxon rank sum exact tests if assumptions unmet) and paired sample t-tests between sessions (Wilcoxon test if assumptions unmet).

Extracting Other Variables

Lotte and Jeunet (2017) suggest that online classification accuracy does not provide an adequate measure of participant performance and provide MATLAB scripts to calculate other variables. Their MATLAB code was used to calculate both stability and distinctiveness of the EEG signal.

For these calculations, the exact same preprocessing of the EEG signal was done on a two second interval just before the trial started. This condition served as resting state for comparison.

Stability

Stability and Distinctiveness have been described in Lotte and Jeunet (2017). We closely followed their implementation in Simon and Ruddy (2022) and again in this study. Both measures are implemented to address shortcomings of online CA measurements by conceptualising the variance within a MI state, and the difference between two MI states Lotte and Jeunet (2017).

We used the following implementation of stability (Lotte & Jeunet, 2017; Simon & Ruddy, 2022):

$$\text{Stability of Task A} = \frac{1}{1 + \sigma_{C^A}} \quad (7)$$

With σ_{C^A} being the standard deviation of the covariance matrix in Riemannian space. The covariance matrix being defined as the average of single trial (i) covariance matrices $C_i = \frac{1}{N_s} X_i^T X_i$ where N_s is the number of samples per trials, X_i is the EEG signal from all channels, and T being transpose (Lotte & Jeunet, 2017). A full explanation of these concepts can be found in Lotte and Jeunet (2017).

We used the following implementation of distinctiveness (Lotte & Jeunet, 2017; Simon & Ruddy, 2022):

$$\frac{\delta(C^A, C^B)}{\frac{1}{2} (\sigma_{C^A} + \sigma_{C^B})} \quad (8)$$

Where the Riemannian distance $\delta(C^A, C^B)$ between the signals of two mental tasks A and B is divided by the averaged covariance matrixes $\left(\frac{1}{2} (\sigma_{C^A} + \sigma_{C^B})\right)$. In this case the two tasks, A and B, would be the two MI conditions of Up and Down.

4.2.4 Results

The results are split into three sections: the analysis of the classification accuracy, stability and distinctiveness, and spectral power. For most analyses the participants were split into a group of successful Brain-Computer Interface (BCI) controllers and unsuccessful noncontrollers. Successful controllers were determined by having a Classification Accuracy (CA) score above 50, averaged over the last two sessions.

A block was deemed to have inappropriate feedback if the confusion matrix of the classification accuracy was completely left or right sided: Containing 20 true positives and 20 false positives or containing 20 false negatives and 20 true negatives. This was the case for 68 (out of 282 trials). As it was impossible to determine post-hoc what led to this inappropriate feedback, the data were not removed. We suspected that some inappropriate blocks were due to fatigue or motivational issues from the participant side. However, we mainly suspect that the CSP and LDA were inadequately calibrated – meaning they were trained on inadequate data or used for too long without during the experiments.

Participants (1, 5, 6, 7, 8, 9) achieved classification accuracies significantly above chance levels in the last session. Due to the preprocessing steps a block averaged 25.65 epochs and had an average of 1.71 interpolated electrodes.

After preprocessing quality indicators such as number of epochs, number of electrodes, number of blocks per session and number of rejected sessions were analyzed. Setting a threshold of accepting trials with nine or more epochs, 18 blocks (of 282) were rejected. Rejecting epochs with less than eleven noninterpolated electrodes, ten additional blocks (28 of 282) were rejected. Excluding sessions with less than two blocks per session and excluding participants with more than one missing session resulted in the rejection of one participant (participant 8).

Overall, in this preprocessed dataset the average number of retained epochs increased by 1.76 and the average of retained noninterpolated electrodes increased by 0.24. Analyzing the CA using the preprocessed data one participant (6) changed group from controller to noncontroller. In total therefore, the preprocessed dataset loses two controllers (6, 8) and gains one noncontroller compared to the raw dataset.

The thresholds of quality indicators (Number of epochs, channels etc.) were chosen to minimize the inclusion of outliers and maximize available data.

Classification Accuracy

A mixed effects linear model was used to analyse classification accuracy. The mixed effects model was created in RStudio (Version 1.4.1106), see figure 4.2. The model predicted CA using the block number as a fixed effect while using block number and individual participants as random effects. The random effects adapted an intercept to individual starting CA and a slope to individual learning rates. Therefore, the fixed effect of block number was accounting for repeated measures or as a variable tracking the amount of training a participant had received. In this model block number was significant (Beta = 0.5, t-Value = 2.6, p-Value = 0.024).

Stability and Distinctiveness

As the calculation of stability and distinctiveness is based on individual trials of EEG signal during blocks, only blocks with more than three trials per condition were accepted. This led to the analysis of a total of 176 trials. Six (3.41%) trials were larger than three times the respective standard deviation. These outliers were removed from further analysis. This resulted in the exclusion of participant 9.

There were more Up (M = 9.42, SD = 3.97) than Down (M = 8.78, SD = 3.86) trials $t(164) = 2.28$, $p = 0.024$.

Stability

A mixed effects model trying to predict the stability of the respective state (Rest, Up and Down) using the block number was created. The model predicted stability using the block number as a fixed effect while also using block number and individual participants as random effects. The random effects adapted an intercept to individual starting stability and a slope to individual learning rates. Therefore, the fixed effect of block number was accounting for repeated measures or as a variable tracking the amount of training a participant had received. Training only improved the stability of the Up state (Beta = 0.0014, t-Value = 2.35, p-Value = 0.045).

A model with the same random effects predicting CA using the stability of the different mental states was not significant.

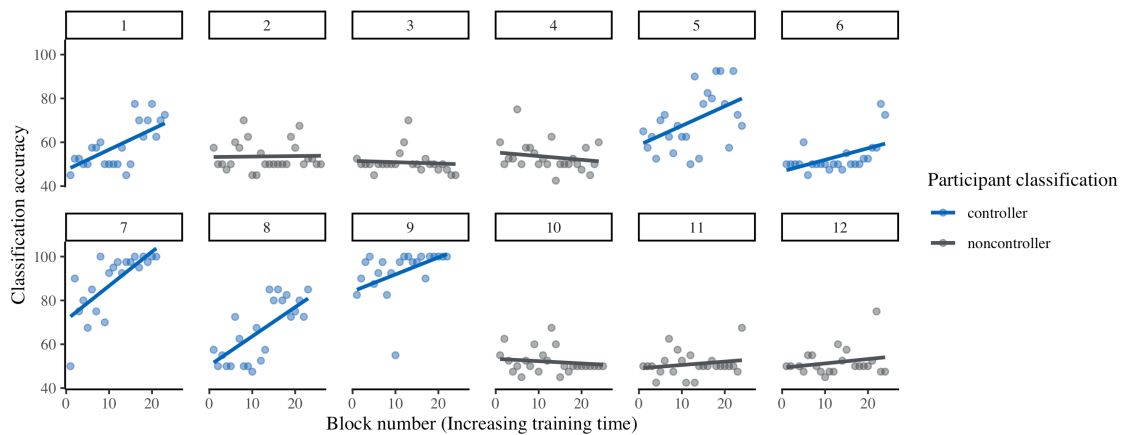


Figure 4.2. Classification Accuracy per Participant. Classification accuracy of individual participants over time. Participants with CA significantly over 50 (controllers) are blue, others (noncontrollers) are grey. A mixed models regression is fitted to each participant with an individual intercept (starting ability) and slope (learning rate).

Distinctiveness

Distinctiveness is measured between two states. Therefore, every value corresponds to the distinctiveness between two states such as Rest vs Up, Rest vs Down, or Up vs Down. A mixed effects model trying to predict the distinctiveness between states using block number was created. The model predicted distinctiveness using the block number as a fixed effect while also using block number and individual participants as random effects. The random effects adapted an intercept to individual starting distinctiveness and a slope to individual learning rates. Therefore, the fixed effect of block number was accounting for repeated measures or as a variable tracking the amount of training a participant had received.

The distinctiveness of the Up state compared to the Down state increased significantly over time (Beta = 0.0083, t-Value = 2.53, p-Value = 0.032). A model with the same random effects predicting CA using the distinctiveness between the Up and Down state was significant (Beta = 8.27, t-Value = 2.27, p-Value = 0.025).

In a model with the same random effects using both block number as well as distinctiveness between the Up and Down state as fixed effects the distinctiveness between the Up and Down state was significant (Beta = 7.6, t-Value = 2.07, p-Value = 0.04), while block number was not (Beta = 0.35, t-Value = 1.76, p-Value = 0.11).

Spectral Band Power

The analyses investigated differences between the Up and Down state and between controllers versus noncontrollers. To compare the effects of training, the power values were

taken from the first and last session. The power of Down trials was subtracted from the power of Up trials and groups of controllers and noncontrollers were compared. The power differences between individual conditions (Up and Down) were also investigated. The data were compared using independent sample t-tests between groups (or Wilcoxon rank sum exact tests) and paired sample t-tests between sessions (or Wilcoxon tests).

FC3 Alpha Power

The power between Up and Down trials was significantly different between the groups (controllers' median = -0.11, and noncontrollers' median = 0.026) at the start of the experiment (session 1, $W = 44$) $p = 0.001$. By the end of the experiment (session 6) the power difference within controllers between the starting ($M = -0.2$, $SD = 0.24$) and ending session ($M = -0.48$, $SD = 0.44$) had significantly changed $t(12) = 3.65$, $p = 0.0033$, while noncontrollers had not changed (start median = 0.026, end median = -0.08, $V = 87$, $p = 0.14$). Comparing the groups at the end of the experiment also yields a significant change (controllers' median = -0.46, noncontrollers' median = -0.08, $W = 51$) $p = 0.013$.

Comparing trial conditions controllers show a significant power difference at the first session (Up $M = 1.19$, $SD = 0.69$, Down $M = 1.38$, $SD = 0.87$, $t(12) = -2.99$, $p = 0.011$), and the last session (Up $M = 1.01$, $SD = 0.61$, Down $M = 1.49$, $SD = 0.75$) $t(17) = -3.73$, $p = 0.0015$.

FC3 Beta Power

The beta power follows the same pattern as alpha power.

The power between Up and Down trials was significantly different between the groups (controllers' median = -0.06, and noncontrollers' median = 0.016) at the start of the experiment (session 1) $W = 69$, $p = 0.037$. By the end of the experiment (session 6) the power difference within controllers (start $M = -0.057$, $SD = 0.14$, end $M = -0.22$, $SD = 0.17$) had significantly changed $t(12) = 3.10$, $p = 0.0091$, while noncontrollers had not changed (start median = 0.016, end median = -0.022, $V = 57$, $p = 0.81$). Comparing the groups at the end of the experiment yields a significant change (controllers' median = -0.21, noncontrollers' median = -0.022, $W = 43$) $p = 0.00039$.

Looking at the trial types, controllers show a significant power difference in the last session (Up: $M = 0.75$, $SD = 0.42$, Down: $M = 0.96$, $SD = 0.43$), $t(13) = -4.83$, $p = 0.00033$.

Lateralization of Power

Lateralization was operationalized as the difference in power measured at the FC3 and the FC4 electrode (FC3 – FC4). This difference was compared in both trial conditions within controllers. As they were unable to control the BCI noncontrollers were not analysed.

In the alpha and beta power band, no significant lateralization differences were found, see figure 4.3. Instead, comparisons showed that the FC4 electrode mirrored effects observed at the FC3 electrode.

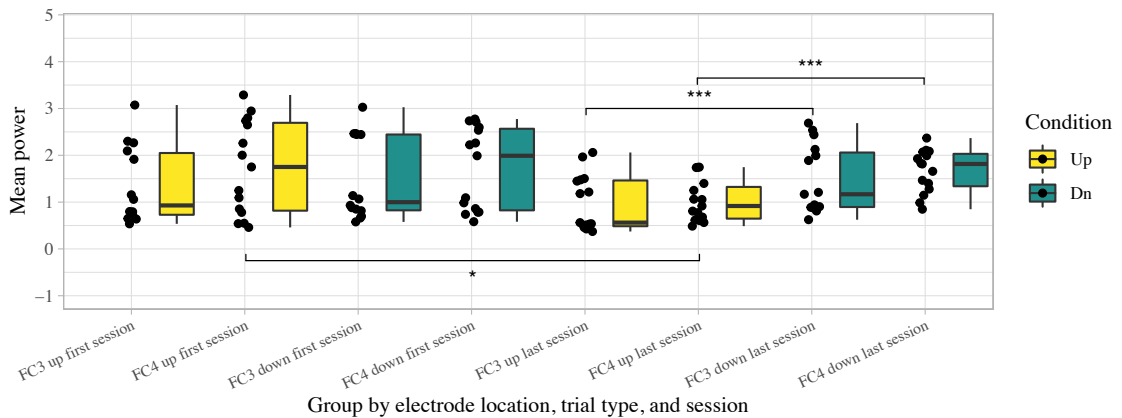


Figure 4.3. Mean Alpha Power Between Electrodes, Conditions and MI. Comparison of mean alpha power between trial conditions, sessions, and electrode locations. Significant effect at the FC3 between the Up and Down trials of the last session. Instead of lateralization (different behaviour of the FC4 electrode) this effect is mirrored at the FC4 location, with the FC4 location showing a significant difference between the first Up session and last Up session.

The difference in alpha power in Up trials compared to Down trials in the last session is also found at the FC4 electrode (M Up = 1.03 SD = 0.45, M Down = 1.66 SD = 0.46) $t(14) = -5.34$ $p = 0.0001$. Another significant comparison is between the first and last Up sessions' power at FC4 (median first = 1.63, median last = 1.06, $V = 68$) $p = 0.021$, but not FC3.

Similarly, the difference in beta power in Up trials compared to Down trials in the last session also found at the the FC4 electrode (M Up = 0.72 SD = 0.41, M Down = 0.94) $t(14) = -5.18$ $p = 0.00014$. Another significant comparison is between the first and last Up sessions' power at FC4 (M first = 1.15, SD = 0.65, M last = 0.76, SD = 0.27) $t(11) = 2.71$, $p = 0.021$, but not FC3.

4.2.5 Discussion

We show that attaining control over a BCI is possible within 6 days using a 16 dry electrode, wireless, wearable EEG system. We demonstrate that this change is not only

confined to Classification Accuracy (CA) but that over time the controllers produced more stable Up and more distinction between the Up and Down states. The metric of distinctiveness was more strongly associated with CA than training.

The alpha (8-13Hz) and beta (14-30Hz) power measured at the FC3 and FC4 electrode significantly differed between and within controllers but not within noncontrollers. The lateralization of activity was not significant. The study results suggest that dry, wireless, wearable setups are feasible for BCI use at home and that it is possible to remotely instruct users in setting up and using a BCI. While individual attributes of feasibility for an at home motor imagery BCI have been investigated – such as low electrode count BCIs (Memon et al., 2018) or restoring function (Lin et al., 2019) – we have combined multiple factors; A convenient setup with a potential clinical protocol for neurorehabilitation at home.

Procedure

The described online procedure with two video calls was successful. Participants rarely needed additional video calls, and most were able to set up and use the BCI without complications. However, some aspects could be improved.

While participants were encouraged to attempt to set up the BCI without help, most attempted to do this during or shortly before the first video-call – meaning most were assisted during setup. However, after the second video call participants rarely needed additional help. Sometimes the participants encountered issues that were not able to be resolved easily by themselves or the experimenters, so for future studies it may be advisable to have an incident reporting form to compile errors. Overall participants reported the set up being complex but manageable due to the instructions made available as documents and during video calls.

Participants were very accepting of the experiment and interested in both the experiment and the outcome. After the completion of the experiment, we offered another video call explaining the aim of the experiment in detail and outlining each step of the experiment in as much detail as possible. However, many participants noted that the experiment was too long, and that by the end of the session, they were not able to concentrate optimally. Initial head measurements were not always adequate, and sometimes the EEG cap did not fit the participants on the first try. Some participants also noted that wearing the EEG cap for longer than 30 minutes was uncomfortable. This may have been exacerbated by the limited choice in cap sizes.

Classification Accuracy

To ensure that the effect of training on CA was not due to biased data selection, we investigated another dataset that only used blocks where feedback was appropriate. With this dataset the model for block number remained significant ($\text{Beta} = 0.49$, $t\text{-Value} = 2.54$, $p\text{-Value} = 0.028$).

Power

The results of this feasibility study – A bilateral reduction of both alpha and beta power in the comparison between Up and Down conditions – are difficult to compare to other literature. The Up condition is comparable to (Neuper et al., 2009) who use an almost identical setup but slightly different task for a part of their experiment. They report a bilateral reduction of alpha and beta power during right-handed imagery with feedback. However, using the same task instructions with a different setup (Ruddy et al., 2018) found a left lateralized lower power during Up compared to Down trials with an additional beta power

reduction in a subgroup of the control group. Analysing each trial condition separately does yield more comparable literature for the Up state, but not for the Down state.

Motor imagery of the right hand is likely left lateralized (Stinear et al., 2006b), but multiple factors are known to influence the laterality of motor imagery. The Up condition resembles a long motor imagery trial with abstract feedback – both favouring bilateral activation (Nam et al., 2011; Neuper et al., 2009). However, the complex imagery in the instructions should increase laterality (Sabaté et al., 2004). Additionally, compared to many BCI setups this feasibility study does not include haptic feedback, which may lead to increased lateralization as seen in some experiments (Kaiser et al., 2011; Pfurtscheller et al., 2003; Pfurtscheller & Neuper, 1997).

The Down condition instructions are similar to manipulations of body ownership (Majid et al., 2015). These manipulations have been studied within the context of the rubber hand illusion (Golaszewski et al., 2021). However, comparisons are difficult as body ownership is usually manipulated with visual and tactile feedback and for the purpose of increased ownership over a fake limb.

The task instructions have multiple possible patterns of activation described in literature. Therefore, tailoring BCI components to these tasks is difficult. More specific knowledge of the frequency and localisation of power could increase the learning rate and successful use of this BCI.

Future Outlook

For BCIs to reliably find their way into medical practice they need to become more favourable than alternatives – which means improving scalability and clinical justifiability (Simon et al., 2021). This study increases both by suggesting that independent BCI use at home is feasible. However, there are still challenges to overcome.

Overall success and learning rates need to improve. This might be possible by improving signal quality and signal processing. Improving the protocol instructions might lead to less signal quality losses due to human error and using signal processing methods designed for a specific rehabilitation procedure might increase the computational capabilities of a BCI. Lastly, an approach centred on usability is also likely to improve success and learning rates (Jeunet et al., 2020).

Conclusion

This study presents results indicating that learning to control a BCI using a convenient, dry, wireless, 16 electrode EEG setup is possible. Participants were able to improve their CA scores during the experiment, and EEG analysis shows that controllers successfully changed their patterns of EEG activity whereas noncontrollers did not change. Data collection was done by participants at home and administrator instructions were done online, enhancing credibility of BCI therapy approaches at home.

4.2.6 Author Contributions

CS wrote the first draft and edited the final draft. KR and CS conceptualised the idea for the manuscript. KR and CS worked on subsequent drafts of the manuscript editing and providing feedback. All authors reviewed the final version and were involved in discussions throughout.

CRedit Contributions

Conceptualization: C.S. and K.R.; Data curation: C.S.; Formal analysis: C.S.; Funding acquisition: K.R.; Investigation: C.S. and K.R.; Methodology: C.S. and K.R.; Project administration: C.S. and K.R.; Resources: C.S. and K.R.; Software: C.S. and K.R.; Supervision: C.S. and K.R.; Validation: K.R.; Visualization: C.S.; Writing – original draft: C.S.; Writing - review & editing: C.S. and K.R.

5. Contemporary Considerations for BCI Design in Neurorehabilitation

5.1 Chapter Introduction

Following the impact of the published and presented work described in the previous three chapters, and having been nominated for the BCI awards 2022 we were invited to write a book chapter for a publication accompanying the awards. We summarised and generalised the knowledge gained from our research programme thus far, focusing on why our approach is different from what is currently described in Brain-Computer Interface (BCI) literature, along with the medical implications.

5.2 Keeping Our Eyes on the Prize; Are we Losing Sight of the ‘Why’ in BCI for Neurorehabilitation?

“Technology is the answer. But what was the question?”

Cedric Price (1979)

Studies using Brain-Computer Interface (BCIs) based upon noninvasive, scalp recorded Electroencephalography (EEG) have consistently demonstrated utility, both as scientific tools for neuromodulation and for clinical neurorehabilitation purposes. They are particularly appealing in clinical contexts where physical movement is impaired, for instance following stroke. Using a BCI where on-screen avatars are driven by neural activation in motor regions encourages the patient to engage in imagined or attempted movements. By providing tangible visual feedback and rewarding desirable neural features, activity in motor pathways is maintained. This may promote use-dependant plasticity and rewiring for recovery of function. However, clinical adoption of the approach has been limited. This is due mainly to difficulties with implementation in nonresearch settings, as training to achieve neural control of the BCI requires lengthy sessions over multiple days or weeks (Simon et al., 2021). Human participants typically use imagined movements or scenarios to achieve BCI control, that are associated with distinct scalp-detectable patterns of neural activity. The lack of spatial sensitivity of EEG to distinguish between different imagined (or even executed) movements from different muscles, makes the task of decoding and classifying the neural activity challenging, especially in the early days of training. Transcranial

Magnetic Stimulation Neurofeedback (TMS-NF) is a novel approach to provide richer, more muscle specific feedback in a BCI (Ruddy et al., 2018). In this closed loop system, the size of Motor Evoked Potentials (MEPs) in response to TMS are incorporated into the NF signal, such that users can learn how to facilitate or inhibit their brain-muscle pathways by engaging mental strategies to make the MEP larger or smaller (Figure 5.1). This typically results in a much faster (i.e. within one training session) grasp of how to execute distinct mental strategies in order to control the game, compared to those reported for EEG BCI. This also produces more spatially distinct and temporally stable neural signatures when the same mental strategies are later executed during simultaneous EEG recording (Mihelj et al., 2021). TMS-NF would not be a practical long-term approach for neurorehabilitation due to the expense and size of the required apparatus, but if it could be used in the early phase of training while still in hospital, the learned strategies may transfer well to an EEG BCI that could be used by patients in their own homes (Simon & Ruddy, 2022, Figure 1).

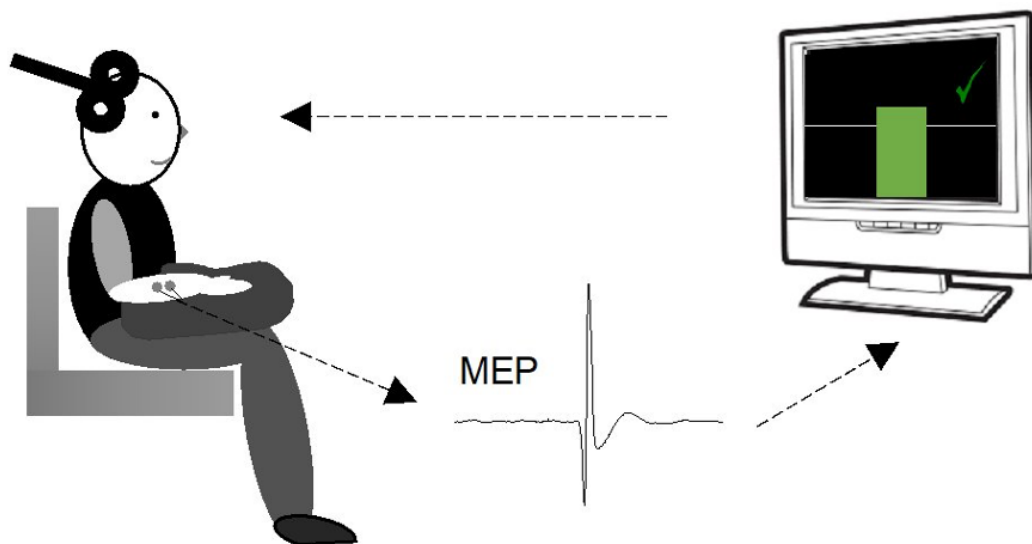


Figure 5.1. TMS-NF Setup. TMS Neurofeedback protocol used in Ruddy et al. (2018). The peak-peak amplitude of the motor evoked potential (MEP) following TMS is calculated in real-time and displayed to the participant. The height of the rectangle on screen represents the amplitude of their MEP. In UP training, participants are rewarded for exceeding their baseline amplitude (white horizontal line). In DOWN trials, they are rewarded for keeping amplitudes small, below the baseline. In a successful trial, the bar turns green at the moment of exceeding baseline and a positive soundbite is played. In an unsuccessful trial, the bar turns red, and a discouraging soundbite is played.

Stroke is one of the leading causes of death and disability worldwide, and incidence is rising at an alarming rate (McElwaine et al., 2016). There have been substantial improvements to emergency stroke care, with advances in thrombolytic therapy leading to a

25% reduction in stroke related deaths (National Clinical Programme for Rehabilitation Medicine, 2018). While this is undoubtedly a positive advance, the more worrying implication is that there is now an increasing number of survivors left with significant disabilities as a result of the brain damage, preventing them from leading independent lives. A major unsolved problem is getting the paralysed upper limb functional again. This is one of the key obstacles preventing independence in stroke survivors, as without the use of one limb, they are unable to perform the most basic activities of daily living, such as dressing, preparing food, and personal care. As the brain-muscle pathways that previously controlled the upper limb are permanently damaged, recovery of function may only occur due to growth and development of new neural connections. In this field of research, novel, theory driven, innovative methods to promote this re-wiring of neural pathways are urgently needed.

BCIs take a wide range of different formats, providing user feedback from a variety of different neural signals. Some constitute assistive technology, circumventing the damaged nervous system to translate an intention to move into remote control of an external device. Others work upon harnessing the power of motor imagery to activate brain circuits involved in movement in order to promote an optimal neurochemical environment for neuroplastic processes to occur, and prevent further deterioration of unused pathways. The aforementioned are two very different approaches – one with a focus on compensation, finding a new way to achieve the desired task, and the other with a focus on restitution – regaining the capacity for the brain to control the appropriate muscles by reliance on rewiring and forming new neural connections to take on the function of the damaged neurons. Here we aim to more specifically demarcate the clinical instances in which the benefits of BCI are likely to supersede those of other approaches, with a view towards equipping researchers (and ultimately clinicians) with tools to choose the right BCI approach at the right time for the right patients.

First and foremost, we need to develop a clearer understanding of the underlying mechanisms – the neural elements that are being altered by the BCI – in order to provide clear medical justification, and a more standardised, scalable application of brain-based physiotherapy. 26/10/2023 14:58:00There is a wide heterogeneity in BCI design, slowing its clinical deployment (Baniqued et al., 2021). Yet, a more pressing challenge is our limited understanding of the mechanisms that drive BCI control and BCI initiated recovery processes (Simon et al., 2021).

The myriad of proteins, cells, synapses, structural and functional networks that make up the brain, and the way they interact to produce measurable signals at the scalp, are yet to

be consolidated into a comprehensive framework of brain health (Bassetti et al., 2022; Cattaneo et al., 2018; Chen et al., 2022). Instead of resorting to increasingly complex and expensive technology (Saibene et al., 2023), it is crucial to devise innovative but minimalist ways to interact with the injured brain.

5.2.1 Identification of Appropriate Target Neural Signal

Any form of brain signal, ranging from protein-level fluctuations to network activities and oscillations, can be utilized for BCI. It is also valuable to consider the potential applications where brain-derived signals, such as motor evoked potentials from electromyography (Mihelj et al., 2021; Ruddy et al., 2018), can be leveraged as inputs to inform a BCI (Cruz et al., 2019; Zimmermann et al., 2011). Here, we will focus predominantly on EEG BCI, as, in practical terms, the majority of BCIs capture brain activity through EEG (Aggarwal & Chugh, 2022).

For a BCI with a goal involving neurorehabilitation, the primary objective is to restore functional movement in the user. This distinguishes it from efforts to achieve the best possible control over the avatars/actuators in the BCI which is often the focus from a bioengineering perspective. Rehabilitative BCI can target at least three separate dimensions for brain signal analysis: The signal will originate from a specific location, it will exhibit a distinct temporal development, and it will propagate itself spatiotemporally with an oscillation (Cohen, 2014). Researchers have successfully targeted all three dimensions for controlling BCI (Abiri et al., 2019; Aggarwal & Chugh, 2022; Saibene et al., 2023; Xie & Oniga, 2020). In the spatial dimension, post-stroke BCI for rehabilitation may bring about functional change by enhancing activity dependant plasticity in motor regions (Overman & Carmichael, 2014; Stinear et al., 2020). In the temporal dimension the aim may be to augment conduction speed between the brain's motor areas (Beuter et al., 2018, 2020). From a neural oscillation perspective, a target mechanism may be to bias neural oscillatory behaviour towards more 'normal' patterns than those typically detected from the post-stroke brain (Leonardi et al., 2022).

In a rehabilitation context there are further issues to consider that do not present when testing with healthy controls: Firstly, signals recorded from the injured brain are often a complex and idiosyncratic, and secondly, the chosen brain signal can be anticipated to evolve as the patient recovers. To deal with these complexities we suggest a multi-phase BCI approach. Akin to physiotherapy a multi-phase BCI may start targeting a specific brain signal during the acute phase of a brain injury, and progressively address patient needs until another BCI phase begins that may target another brain signal during the chronic phase of a brain

injury. Ultimately, the best method to accomplish this depends on the specific rehabilitative objectives pursued.

5.2.2 Identifying Therapeutic Goals

The intrinsic advantage of BCIs over alternate rehabilitation strategies is that they work even when output at the behavioural level is nonexistent. Patients exhibiting minimal or no residual limb movement after a stroke cannot partake in gold standard physiotherapy (Kwakkel et al., 2015), but might still demonstrate brain activity patterns when attempting to move the impaired limb. These patterns can be targeted to enhance recovery (Simon et al., 2021). However, the role of BCIs should evolve once behavioural output is available. Uncertainty about how BCI can engage neural recovery mechanisms in the brain mean that standalone BCI therapy is unlikely to produce better results than conventional physiotherapy in cases where behavioural output is available. In this context the role of the BCI may transition from being the primary feedback and training mechanism for a patient to take on a more complementary role in the recovery process.

5.2.3 Identifying the Most Appropriate Technology to Improve Function

In BCI research for neurorehabilitation, the primary objective is to facilitate neural recovery. Achieving accurate BCI control is secondary. As such, rehabilitative BCI is not so much a competition to employ the most cutting-edge methods for the best possible BCI control, but rather a race to identify the most clinically relevant aspects of healthy or injured brains.

So-called black box machine learning methods, where the result is uninterpretable to humans, are not recommended as BCI components (Sau Wai Tung et al., 2013). While they may have advantages over more traditional methods, many require large datasets, which are generally unsustainable in both the academic and medical fields. Furthermore, since researchers or clinicians are not able to oversee the results, these methods carry risks such as overfitting and negative adaptation, rendering them ethically questionable as well as impractical (Adadi & Berrada, 2018; Pinto et al., 2022). As a result, theory-led methods are preferable.

In situations where there are few or no constraints imposed on the data by the chosen brain signal or the rehabilitative objective, unbiased data processing methods can provide benefits. Unbiased methods take all available data at face value and either do not attribute

weights on that input or determine for themselves what is important and what is not, without referring to predetermined goals, hypotheses, or other inputs from researchers. They can offer greater flexibility in identifying optimal solutions for BCI control, including adaptations to signal changes over time, adaptations to patient idiosyncrasies, and the exploration of potential solutions not previously considered by researchers. However, unbiased processes may lack the necessary methodological power to validate BCI mechanisms and are potentially more susceptible to noise.

Based on the qualities of the brain signal and the requirements dictated by the rehabilitative objective, most BCI designs will gravitate towards a specific technological solution. If the qualities and the rehabilitative objective alter during the rehabilitation process, the technology employed should adapt correspondingly. While this multimodal approach may initially appear more complex, it is likely to yield superior results and more standardized protocols.

5.2.4 Thinking Outside the (Black) Box

The presented steps are intended to help BCI design by encouraging innovative solutions. For example, multimodal and multiphasic BCIs may also incorporate phases or modes where the brain is not the signal generator but rather a signal processor. The brain could be the receiver of a stimulus such as transcranial magnetic stimulation or a sensory stimulus. This could enhance the specificity of the generated brain signal along various dimensions, making it easier to design a rehabilitative objective around it.

For this purpose, we have developed a multimodal and multiphasic BCI using a TMS-BCI in the initial phase and an EEG BCI in the subsequent phase. Preliminary results suggest that the multimodal approach was successful in facilitating BCI control compared to a unimodal, single-phase group (2-Phase BCI).

Alternatively, the brain signal could be augmented with additional behavioural data. For example, wearable devices could provide a BCI with an additional signal dimension, enabling the BCI to make more complex decisions. Such multimodal and multiphasic approaches may allow for the targeting of different aspects of rehabilitation during the same treatment or the application of different BCI strengths simultaneously.

A multiphasic approach to rehabilitative BCI could also contribute to its success. To achieve success, BCIs must not only be mechanistically sound and effective but also scalable on a grand scale, ideally allowing patients to use them with minimal clinical supervision. This has yet to be fully realized as researchers continue to strive to find simple and effective

solutions. However, a multiphasic approach may facilitate BCIs to have a short, intensive acute care phase, followed by a long-term easy solution for at-home use.

In an experiment we designed (Simon & Ruddy, 2022), we developed a clinically relevant but easy to use BCI setup. This setup was dispatched to participants' homes, and they were instructed to set it up and train using online guidance. The results affirmed the feasibility of this approach.

5.2.5 Conclusion

Brain-Computer Interfaces (BCI) continue to gain momentum as an appealing method for restoring motor function in a neurorehabilitation context. However, we must not be seduced by the allure of cutting edge technology at the expense of targeting the specific neurophysiological features that are most likely to drive recovery. At the most basic mechanistic level, the majority of BCIs are driven by neural signals generated by imagination of movement. We need to revisit the question: Could motor imagery alone achieve the same outcomes, or what is the added clinical benefit of the BCI? Accordingly, what is the minimum required intervention using BCI (in terms of time and hardware) to establish a habit of good quality motor imagery that could then sustain rehabilitation without the technology? Motor imagery is free, available to every person and at any time. Using technology to harness its virtues while not compromising its simplicity is the ultimate challenge for the field.

5.2.6 CRediT Contributions

Conceptualization: C.S. and K.R.; Funding acquisition: K.R.; Supervision: K.R.; Writing – original draft: C.S.; Writing - review & editing: K.R.

6. Applications of BCI for Rare Diseases and Disorders

6.1 Chapter Introduction

Our work has so far mostly dealt with the consequences of stroke. However, the ability of BCI to connect brain and body has potential for any disease or disorder where the biological brain and body pathway are not working optimally. In this chapter we explored the possibilities of using BCI to help people with Body Integrity Dysphoria (BID) in collaboration with Ms Chakraborty and international colleagues from the team where I completed my master's thesis.

6.2 Could Brain–Computer Interface Be a New Therapeutic Approach for Body Integrity Dysphoria?

6.2.1 Abstract

Patients suffering from Body Integrity Dysphoria (BID) desire to become disabled, arising from a mismatch between the desired body and the physical body. We focus here on the most common variant, characterized by the desire for amputation of a healthy limb. In most reported cases, amputation of the rejected limb entirely alleviates the distress of the condition and engenders substantial improvement in quality of life. Since BID can lead to life-long suffering, it is essential to identify an effective form of treatment that causes the least amount of alteration to the person's anatomical structure and functionality. Treatment methods involving medications, psychotherapy, and vestibular stimulation have proven largely ineffective. In this hypothesis article, we briefly discuss the characteristics, etiology, and current treatment options available for BID before highlighting the need for new, theory driven approaches. Drawing on recent findings relating to functional and structural brain correlates of BID, we introduce the idea of Brain-Computer Interface (BCI)/neurofeedback approaches to target altered patterns of brain activity, promote re-ownership of the limb, and/or attenuate stress and negativity associated with the altered body representation.

6.2.2 Introduction

For most people, the thought of becoming physically disabled is troubling and something to be avoided at all costs. For patients suffering from body integrity dysphoria (BID) however, the desire to become disabled is characteristic, arising from a mismatch between the desired body and the physical body in terms of functionality or shape. In its most common variant, i.e., the desire for amputation, on which we focus here, the defining

features include an intense feeling of inappropriateness concerning their current body configuration. BID patients desire to become physically disabled in order to feel “complete” and become the person they envisage themselves to be (First, 2005; White, 2014). Ironically, achieving this completeness often involves desiring to amputate an otherwise healthy and fully functioning limb, raising many ethical and medical considerations. In most reported cases, amputation of the rejected limb entirely alleviates the distress of the condition and engenders substantial improvement in quality of life (Noll & Kasten, 2014). Individuals with BID are non-psychotic and do not otherwise present major psychiatric or neurological disorders. Sufferers often describe themselves as being trans-abled, drawing a parallel with transgendered individuals.

Since BID can lead to life-long suffering, it is essential to identify an effective form of treatment that causes the least amount of alteration to the person’s anatomical structure and functionality. Current forms of treatment such as amputation are extreme and lead to a permanent disability and presumably dependence on carers for activities of daily living. Other treatment methods such as medications, psychotherapy, and vestibular stimulation have proven largely ineffective (Ryan, 2008; Blom et al., 2012). In this article, we briefly discuss the characteristics, etiology, and current treatment options available for BID before highlighting the need for new, theory driven approaches. Drawing on recent findings relating to functional and structural brain correlates of BID, we introduce the idea of Brain-Computer Interface/neurofeedback approaches to target altered patterns of brain activity, promote re-ownership of the limb, and/or attenuate stress and negativity associated with the altered body representation.

6.2.3 Characteristics and Aetiology of BID

In order to provide an in-depth evaluation of BID and its manifestations, it may be helpful to first differentiate from other similar conditions such as body dysmorphic disorder (BDD) and somatoparaphrenia, among others (Table 6.1). BID shares similarities with BDD, because individuals with BID are also found to be preoccupied with certain aspects of their body (Grochowski et al., 2018). However, there are some key differences. The dysphoria experienced by individuals with BDD is derived from their concerns over how a part of their body physically appears to others (First & Fisher, 2012). Specifically, individuals with BDD believe that some aspect of their physical appearance is defective and as such is a source of intense embarrassment, shame, and low self-esteem.

Somatoparaphrenia (SPP) is characterized by a denial of ownership combined with confabulatory signs (Gerstmann, 1942). It affects predominantly the left side of the body

(McGeoch et al., 2011). Two main clinical features separate it from BID. First, while BID likely develops in childhood and may be related to amputation-related experiences (Barrow & Oyeboode, 2019), SPP is an acquired condition in response to brain damage, typically of the right hemisphere, that typically also causes contralateral left limb paralysis. Moreover, while BID individuals recognize the unwanted limb as part of their current but undesired body, SPP is a disownership disorder, primarily characterized by a positive component, reflecting a “distorted mental representation of reality” (Bottini et al., 2009, p. 589), that is the delusional misattribution of the paralyzed limb to someone else (Gerstmann, 1942; Romano et al., 2014; Saetta et al., 2020). Another related clinical condition is asomatognosia, which is caused by a similar (more often a right lateral) cerebral lesion as SPP. In this condition, however, the limb is felt as nonexistent rather than disowned (Critchley, 1954; Saetta et al., 2021).

Other conditions that have been discussed in relation to BID due to common features are alien hand syndrome and anarchic hand syndrome. For patients with alien hand syndrome, the motor control of the affected limb is disinhibited, and the limb executes smooth but nonvolitional reflexive movements of grasping and compulsive manipulation of tools in the environment, i.e., utilization behaviour (Doody & Jankovic, 1992; Feinberg et al., 1992). Typical damaged brain areas include fronto-mesial and anterior cingulate areas as well as the anterior corpus callosum (Feinberg et al., 1992). Anarchic hand syndrome (AHS) is a related disorder, which ensues from a relatively circumscribed lesion of the anterior corpus callosum (Feinberg et al., 1992, but see Pacella et al., 2021 for a more recent account). In this syndrome, the individuals’ hands move outside of the volitional control as well (Sala, 2009; Jenkinson et al., 2015); however, intermanual conflicts are more typical than utilization behaviour.

Gender incongruity/dysphoria (GD) is a condition in which there is a conflict between aspects of the physical body (sex) and the desired body, often leading to sex-reassignment through surgery and lifelong hormonal treatment. Before the release of the DSM-V, GD was referred to as “Gender Identity Disorder.” The GD diagnosis separates this condition from sexual dysfunctions while framing it in terms of incongruence between phenomenological and assigned gender, instead of “cross-gender” identification disorder. Analogously, while BID has been previously defined as “Body Integrity Identity disorder” (First, 2005), the BID nomenclature focuses on the incongruence between the functionality and/or morphology of the current and assigned body. Conducting a direct comparison between attitudes of those with BID and GD (Ostgathe et al., 2014) revealed that both share

similar onset (early childhood or adolescence), discontent in the individual with their identity, perceived reduction of desire post-surgical intervention, or through mimicking of desired identity such as cross-dressing in Gender Dysphoria and pretending of disability by using wheelchairs or crutches in BID. Comorbidity of BID with Gender Identity Dysphoria (GID) has been described in 19% of the cases (First, 2005; Lawrence, 2010). The groups differed, however, in the intensity of their aversion toward the unwanted body parts, with Gender Dysphoric individuals expressing an intense rejection or hatred, whereas BID individuals exhibited an indifference toward the limb. Remarkably, in both BID and GID individuals, the examination of reality is preserved in that the discomfort sensations are “illusions, not delusions” (Garcia-Falgueras, 2014; p. 161), and do not typically persist after surgical intervention (Garcia-Falgueras, 2014).

Table 6.1*Taxonomy of Disorders Related to Body Integrity Dysphoria*

Disorder	Key Features		Implicated Brain Regions
Body integrity dysphoria (BID)	Desire to become physically disabled (e.g., blind, paralyzed, and amputated). Amputation variant includes (1) dysphoric feelings about a typical body morphology and/or functionality originating from a mismatch between the mental body image, i.e., the conscious representation of body size and shape, and the perceived body (Saetta et al., 2020), (2) lack of ownership over a limb, (3) excessive attention devoted to the limb, and (4) intact insight regarding the condition and its unusual nature (Giummarra et al., 2011). Most often left limb or both limbs (First, 2005). Typically observed in BID are also paraphilic interests, such as preferences for an amputated sexual partner and sexual arousal by the idea of being an amputee (Blom et al., 2017)		- Right superior parietal lobule (Ramachandran et al., 2009; McGeoch et al., 2011; Hilti et al., 2013; Saetta et al., 2020) - Left premotor ventral cortex Saetta et al., 2020; van Dijk et al., 2013; Blom et al., 2016) - Right primary somatosensory and motor cortices (Hilti et al., 2013; Saetta et al., 2020) - Bilateral nucleus caudatus and putamen (Hänggi et al., 2016)
	Shared Features with BID	Features Distinct from BID	
Body dysmorphic disorder (BDD)	Preoccupation with certain aspects of the body (Grochowski et al., 2018)	Dysphoria experienced by individuals with BDD is derived from their concerns over how a part of their body physically appears to others (First & Fisher, 2012). Specifically, individuals with BDD believe that some aspect of their physical appearance is defective and as such is a source of intense embarrassment, shame, and low self-esteem	- Reduced volume of the orbito-frontal cortex and the anterior cingulate, and larger volume of the thalamus (Atmaca et al., 2010) - Frontostriatal and temporoparietal occipital circuits for visual-spatial processing (Mufaddel et al., 2013)
Somatoparaphrenia	Denial of ownership over a limb (Gerstmann, 1942), and more often concerns the left body part (McGeoch et al., 2011).	Acquired condition in response to brain damage, typically right hemisphere, which may cause left limb paralysis. Characterized by delusional misattribution of the paralyzed limb to someone else (Gerstmann, 1942; Romano & Maravita, 2019)	- Right hemisphere subcortical white matter (Gandola et al., 2012; Moro et al., 2016) - Middle and inferior right frontal gyrus (Gandola et al., 2012) - Right hippocampus and amygdala (Gandola et al., 2012; Romano et al., 2014)

Asomatognosia	Disruption to right hemisphere influencing limb awareness Phenomenological similarities in 23% of the cases (Blanke & Metzinger, 2009)	Feeling that parts of the body are missing or have disappeared from corporal awareness (Arzy et al., 2006; Saelta et al., 2020)	Right temporo-parietal lobe, including the right superior parietal lobule (Saelta et al., 2020)
Alien hand syndrome	Disownership of the limb No traceable underlying cause for the perceived "disembodiment" of the limb (Müller, 2009)	Motor control of the affected limb is disinhibited and moves nonvolitionally	- Primary motor cortex, premotor cortex, precuneus, and right inferior frontal gyrus (Schaefer et al., 2010) - Supplementary motor area, anterior cingulate gyrus, medial prefrontal cortex, and anterior corpus callosum (Feinberg et al., 1992)
Anarchic hand syndrome	Disembodiment of the limb Potential underlying mechanism relating to white matter disconnection and brain lesions (Pacella et al., 2021)	Motor control of the affected limb is disinhibited and moves nonvolitionally No disownership of the limb	- Right inferior parietal lobe (Sala, 2009; Jenkinson et al., 2015) - White matter disconnection of antero-posterior, insular, and interhemispheric networks (Pacella et al., 2021) - Anterior corpus callosum (Feinberg et al., 1992)
Gender incongruity	Conflict between aspects of the physical body (sex and the desired body). Onset (early childhood or adolescence), discontent in the individual with their identity, perceived reduction of desire post-surgical intervention or through mimicking of desired identity Comorbidity of BID with GID has been described in 19% of the cases (First, 2005; Lawrence, 2010) Preserved rationality and distress disappears after surgical intervention (Garcia-Falgueras, 2014)	Focus upon gender rather than upon a limb Intensity of rejection of body parts. Intense hatred in Gender incongruity vs. indifference in BID (Ostgathe et al., 2014)	- Bilateral superior parietal lobule and the primary somatosensory cortex (Lin et al., 2014) - Right insula (Nawata et al., 2010)

Before being recently included in the release of the 11th Revision of the International Classification of Diseases (World Health Organization, 2018) as a “disorder of bodily

distress or bodily experience,” BID has also been interchangeably referred to as xenomelia, apotemnophilia, and body-integrity identity disorder (BIID). While the lowest common denominator between these nomenclatures is the desire for amputation, each of them underlies a specific etiological hypothesis (Sedda & Bottini, 2014).

Apotemnophilia (from ancient Greek “love for amputation”) proposes psychological or psychiatric features related to sexual disturbances to be the centerpiece of the disorder, whereas Xenomelia (from ancient Greek “foreign limb”) put the disorder in the context of a neurological syndrome originating from alterations of the right parietal lobe (McGeoch et al., 2011). A lack of consensus regarding the nomenclature concerned with these conditions and their etiological basis is a topic of ongoing scientific debate.

Major characteristics observed in the amputation variant of BID include (1) dysphoric feelings about a normal body morphology and/or functionality originating from a mismatch between the mental body image, i.e., the conscious representation of body size and shape, and the perceived body (Saetta et al., 2020), (2) lack of ownership over a limb, (3) excessive attention devoted to the limb, and (4) intact insight regarding the condition and its unusual nature (Giummarra et al., 2011). Initial studies on BID have shown that in the case of a desire for amputation, most patients with BID desired amputation of either a left-sided limb or both limbs (First, 2005).

Ostgathe et al. (2014) found that justifications given by those with BID desiring an amputation included triggering childhood events, restoring a congruence between sentiment and body, feeling attracted by people with physical disabilities, a fascination for physically impaired people, as well as the perception of body impairment as a proper way of living. Typically observed in BID are also paraphilic interests, such as preferences for an amputated sexual partner and sexual arousal by the idea of being an amputee (Blom et al., 2017).

6.2.4 Current Approaches to Treatment for BID

Individuals with BID find symptomatic relief by spending substantial amounts of time engaged in so-called pretending behaviours in which they simulate the desired disability (e.g., walking with one’s leg tied back, sitting in a wheelchair, using crutches, First, 2005). It is believed that the pretending behaviour allows for the alignment of the perceived and desired body, reducing distress associated with the mismatch (Saetta et al., 2020).

While this provides some transient relief, the preoccupation with becoming disabled has harmful consequences. Time spent engaging in pretending behaviours interferes with the individual’s productivity, leisure activities, and social functioning. In the case of desired

amputation, pretending behaviours such as daily binding up of the leg can also cause injury over extended periods of time. At the most extreme end of the spectrum, attempts to actually achieve the desired disability can have catastrophic effects on the health of the individual, potentially putting their life at risk.

The only truly satisfying solution for BID sufferers seems to be the amputation of the rejected limb (Noll & Kasten, 2014). In most countries, this course of action will not be considered by physicians (Bayne & Levy, 2005) and is strictly illegal, knowing that it renders the individual with a new and unnecessary lifelong physical disability. Thus, alternative approaches targeting the comorbid anxiety and depression symptoms to reduce distress have been the key focus of treatment to date (e.g., Ryan, 2008; Blom et al., 2012). Psychotherapy has been shown to reduce psychological distress, but concurrently increase the individual's desire for amputation, most likely as a result of intense focus and discussion on the topic of their BID with the therapist (Kröger et al., 2014).

Reports of the effectiveness of pharmacological interventions are very limited, and sample sizes are small, often limited to single case reports. One such case report suggested a reduction of distress in one individual with BID from a course of selective serotonin reuptake inhibitors (SSRIs), and limited effects of psychotherapy (Braam et al., 2006). Another report with 25 BID patients found that 12 had been prescribed medications including antidepressants, neuroleptic drugs, and tranquilizers but with no positive effects on symptoms (Kröger et al., 2014).

Another potential treatment that received prior attention was artificial vestibular stimulation, initially proposed by Ramachandran et al. (2007). As disorders of the bodily self have been linked to the vestibular system for over a century (Bonnier, 2009), artificial vestibular stimulation (caloric or galvanic) has been shown to induce temporary remission of the denial of ownership over a limb in SPP (Lopez, 2013) and other bodily self disorders (Grabherr et al., 2015). Given the above-mentioned parallels between the BID and SPP clinical pictures, vestibular stimulation has been tentatively used to restore limb ownership in BID and thus to reduce the desire for amputation. Despite initial excitement and early adoption of the technique clinically, subsequent research studies have demonstrated that it provides no substantial benefit to individuals with BID (Lenggenhager et al., 2014).

Upon review of the available literature, it is evident that aside from amputation, existing treatment options are largely ineffective, and fresh approaches are required to alleviate distress for those with BID. Furthermore, persisting with the current ineffective treatment programs comes at a financial cost to the patients and healthcare providers.

Amputation, although effective, carries the largest financial burden at an estimated \$24,010 for the procedure itself with inpatient stay, and \$878,927 in associated costs over the lifetime of the amputated individual (United States data according to health insurance claims, Lo et al., 2021). With advances in neuroimaging, the field of BID is set to undergo a paradigm shift, availing of new knowledge on brain structural and functional correlates of the disorder that may provide insights toward potential avenues for treatment exploration. In the next sections, we review recent developments for understanding the neurological mechanisms and consider how these may inform the inception of new theory driven approaches to treatment.

6.2.5 Structural and Functional Connectivity Abnormalities in BID

It has been well established since the mid 20th century that brain lesions in the right parietal lobe lead to altered perception and spatial awareness of body parts (Brain, 1941; Critchley, 1954). This area of the cortex is divided into the superior and inferior parietal lobules (the SPL and the IPL, respectively), with the latter being further subdivided into the supramarginal and angular gyri. Within the last decade, research specifically relating to structural and functional brain abnormalities in BID has emerged. In a recent review of all available neuroimaging evidence since 2012, Fornaro et al. (2020) reported that areas consistently found to be altered in xenomelia include right superior and inferior parietal lobule (rSPL and rIPL), premotor cortex, and right insula. A general pattern across studies emerged that cortical abnormalities tended to be right lateralized, whereas subcortical were left lateralized. Hänggi et al. (2016) specifically focused upon subcortical correlates of BID and found distinctive differences in the shape of several structures involved in somatotopic representation of the leg. Patients had thinner bilateral dorsomedial putamina, left ventromedial caudate nucleus, and left medial pallidum. This was accompanied by thicker bilateral lateral pallida and left frontolateral thalamus.

The aforementioned structural evidence is corroborated by data concerning functional connectivity between brain regions at rest in patients with BID compared to controls. In a relatively large and homogenous sample of BID patients who all desired amputation of the left leg, Saetta et al. (2020) found reduced functional connectivity between the primary sensorimotor area (S1) of the disowned leg and the rest of the brain, and the rSPL with the rest of the brain. Further, gray matter atrophy was evident in the rSPL of BID patients, and the extent of this atrophy correlated with desire for amputation and degree of

engagement in pretending behaviours. These findings led Sætta et al. (2020) to suggest that the distressing feeling of nonacceptance of one's limb in BID might ensue from a discrepancy between preserved projections of somatosensory inputs from the limb to the respective primary cortical areas, and an impaired representation of the body at the highest level of integration, the so-called body image. The authors proposed that only a smooth interplay between sensory and higher level bodily processing may provide the phenomenal experience of a limb belonging to the body and the self. The previously identified neural network for (healthy) body image and ownership has been suggested to span a broad network involving primary sensorimotor, premotor, parietal, insular, and extrastriatal areas (see, e.g., Tsakiris, 2010; Grivaz et al., 2017; Park & Blanke, 2019; Seghezzi et al., 2019; Salvato et al., 2020; for reviews), which were, at least to some extent, also found to be altered in BID (Fornaro et al., 2020).

The presence of underlying neuroanatomical structural abnormalities is evidence that BID has both psychological and neurophysiological components, and that recovery may entail modifying hard wired brain circuitry. However, this does not necessarily imply permanency of the condition as both gray and white matter are highly neuroplastic and structural changes over time can be documented following treatment programs for other neuropsychological disorders including depression (Liu et al., 2015; Bouckaert et al., 2016) and alcohol addiction (Pfefferbaum et al., 2014). By altering functional connectivity using targeted therapeutic interventions to promote desired neural patterns, changes in brain structure often follow over time. For example, using brain-computer interfaces, functional and structural neuroplastic changes can be induced in regions specifically involved in the trained task (Ghaziri et al., 2013; Qian et al., 2018; Marins et al., 2019; Nierhaus et al., 2021; Sampaio-Baptista et al., 2020)..

6.2.6 A New Generation of Research into BID Treatment Using Neurofeedback / Brain-Computer Interface

Brain-computer interfaces and neurofeedback have been used in attempts to tune pathological brain rhythms or patterns of functional connectivity toward more neurotypical patterns (Ros et al., 2014). This is achieved by training individuals to use on-screen (or robotic) feedback of real-time brain activity to self-regulate some aspect of their own neural functioning. While BCI can come in a vast array of different arrangements using different neuroimaging modalities (see Simon et al., 2021 for a review), a commonly used setup involves recording electrical activity from the scalp using electroencephalography (EEG),

and presenting feedback on screen in the form of a computer game. With this type of BCI, different aspects of brain activity can be targeted for modification depending on the method used to extract and present feedback. Positive outcomes using this method for symptom reduction have been demonstrated for ADHD (Qian et al., 2018), post-traumatic stress disorder (PTSD, van der Kolk et al., 2016; Banerjee & Argáez, 2017), and generalized anxiety disorder (GAD, Banerjee & Argáez, 2017; Mennella et al., 2017). In the case of BID, given that distinct functional connectivity patterns are evident involving S1, rSPL, and the rest of the brain (Saetta et al., 2020), there may be value in testing whether targeting this pattern using BCI would result in reduction of distress associated with the desire for amputation. For example, training could provide on-screen feedback in a computer game where rewards are achieved contingent on the strength of connectivity between the aforementioned nodes. Over multiple sessions playing the BCI game, strengthening of connectivity could be hypothesized as participants develop or refine mental strategies, and compared to a control condition where pseudofeedback is delivered. Experimental work would be essential to determine effective mental imagery strategies to recommend for participants (or alternatively whether to allow strategy-free implicit learning), and how many sessions of this type of feedback may be required in order to achieve symptomatic relief.

As Saetta et al. (2020) found that gray matter atrophy in BID correlated with the extent of pretending behaviours (such as binding up the leg and simulating amputation), and desire for amputation, it is also necessary to consider whether brain-related changes are a root cause of the disorder or an effect of the resultant coping behaviours. This too has implications for designing interventions based on structural or functional brain connectivity, as the intervention may be off target if focus is upon an epiphenomenal mechanism arising from the pretending behaviour. Even if this is the case, however, BCI approaches to train functional connectivity patterns may still hold value. The simulation of the amputated state may in itself alter brain connectivity patterns for body representations, which may in turn enhance desire for amputation. BCI could be a means to disrupt this loop and find an alternative (less dangerous) method for symptomatic relief that does not perpetuate the altered activity in the brain's body representation regions.

An alternative but related approach would be to use BCI to implicitly promote re-ownership of the limb and engender more positive associations with it. In a study using a foot-specific variant of the rubber hand illusion, Lenggenhager et al. (2015) demonstrated that individuals with BID behaved similarly to healthy controls in that they could take ownership for a fake rubber foot. When stroked synchronously with their opposite foot, they

reported sensations as if it were part of their own body. This implies that in the unwanted limb, not only are the basic senses functioning normally but also that the integration of visual, tactile, and proprioceptive information and visual capture is broadly intact. Thus, the authors suggested that behavioural training could be useful for those with BID based on the fact that they can feel ownership for an artificial limb and deny it for their own limb. Further, recent data from a small case study with two BID individuals revealed that augmented reality (AR) technology could successfully reduce symptoms by engendering the illusion that the unwanted limb is missing (Turbyne et al., 2021).

Some suggest that the desire to amputate arises from a failure to adequately represent the affected limb in the relevant area of cortex. For example, McGeoch et al. (2015) recorded somatosensory evoked fields using magnetoencephalography (MEG) during tactile stimulation of the feet of BID participants. They observed a significant reduction in activity over rSPL in the patients compared to controls. A BCI approach targeting this aspect of BID may involve Neurofeedback (NF) of Motor Evoked Potentials (MEPs) in response to Magnetic Stimulation (TMS). This TMS-NF approach has been shown to increase (or decrease) corticospinal excitability (responsiveness) of the pathways connecting brain to muscle, depending on whether rewarding feedback was provided for large or small MEPs (Ruddy et al., 2018; Liang et al., 2020). In a BID tailored variant of this, it may be possible to incorporate feedback of MEPs from TMS over motor cortical representations of both limbs into the computer game, but bias the feedback over time to gradually reinforce engagement of the unwanted limb. This could be achieved by providing larger or more frequent rewards for large MEPs from the targeted limb. As self-regulation of MEP amplitude is achieved via engagement of motor imagery strategies, over time this may promote re-integration of the limb into the mental schema of body representation/body image. Reducing the discrepancy between actual and desired body representation may subsequently engender more positive affect toward the unwanted limb, albeit implicitly.

However, the use of BCI to treat BID may pose challenges. At this point we are unsure whether people with BID may even consider this a potentially viable treatment option. This could be circumvented by asking people with BID to participate to learn more about their condition, instead of focusing on a medicalised treatment for something they may not consider a medical condition. Further, since previous therapies have shown little to no effect, and the mechanisms involved in BID are not well understood, there is a risk that BCI based intervention may also have minimal effect. Therefore it is more important to focus on the pursuit of working mechanisms instead of therapeutic benefits. However, since there exists

no current treatment, new ideas are generally welcome while minimising the risk of negative side effects.

6.2.7 Conclusion

As BID is a condition with both psychological and neurophysiological attributes, it is yet unclear whether interventions targeting one of these aspects in isolation can serve any benefit. Further, clarification of the relative mechanistic contribution of body ownership vs. disrupted integration of the limb into the body image will be necessary in order to accurately tailor interventions tackling these facets of the disorder. Upon reviewing the available evidence, what is known to date is that pharmacological and psychological therapies are largely ineffective, and that there are no known neurophysiological interventions that are sufficient to alleviate the suffering associated with the desire for amputation. Amputation itself is an extreme solution raising many ethical issues, and one that will not even be considered by most healthcare providers. Pretending behaviours adopted by BID patients as coping strategies raise health concerns and often have dangerous consequences. Thus, new theory driven approaches that leverage recent advances in neuroimaging and technology are greatly needed in order to improve quality of life for BID patients. Those presented here are intended to stimulate debate among those interested in working toward novel solutions, and may ultimately pave the way for a new dialog on approaches to tackle challenging neuropsychological phenomena.

6.2.8 Author Contributions

Colin Simon (CS) Initiated the collaboration between Stuti Chakraborty (SC), Kathy Ruddy (KR) and Bigna Lenggenhager (BL), Gianluca Satta (GS). SC wrote the first draft and edited the final draft. KR and SC conceptualized the idea for the manuscript. KR, SC, CS, BL, and GS worked on editing subsequent drafts of the manuscript and providing feedback. GS and BL significantly extended the sections of the manuscript. All authors reviewed the final version and were involved in discussions throughout.

CRedit Contributions

Conceptualization: S.C., B.L., G.S., C.S. and K.R.; Funding acquisition: B.L. and K.R.; Methodology: S.C., B.L., G.S., C.S. and K.R.; Project administration: S.C., B.L., G.S. and K.R.; Supervision: B.L. and K.R.; Writing – original draft: S.C., B.L., G.S. and K.R.; Writing - review & editing: S.C., B.L., G.S., C.S. and K.R.

6.2.9 Funding

KR and CS were supported by a grant from the Health Research Board of Ireland (HRB-EIA-2019-003). GS and BL received funding from the Swiss National Science Foundation (PP00P1_170511).

6.2.10 Conflict of Interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

6.2.11 Publisher's Note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

7. Structural Brain Connectivity in Elderly Individuals Relating to Motor Inhibition Networks

7.1 Chapter Introduction

The work presented in the following chapter does not involve a Brain-Computer Interface (BCI) and is not directly related to the work we presented in previous chapters. This is because when I started my work on Transcranial Magnetic Stimulation Neurofeedback (TMS-NF) in early October 2019, the plan involved using TMS-NF in a Randomised Controlled Trial (RCT) located in a hospital to investigate the efficiency and working mechanisms of TMS-NF. Among other hypotheses, we wanted to investigate whether TMS-NF could slow or even prevent the decline of white matter pathways observed after stroke (Pinter et al., 2020). I took on the work presented in the following chapter as a small side project to familiarise myself with Magnetic Resonance Imaging (MRI) data. As the COVID-19 Pandemic wore on and the hospital delayed the start of our RCT, my contributions to the MRI project grew in importance.

Therefore, this work focuses on MRI data to connect real world falls with the structural integrity of white matter pathways that connect the motor inhibitory networks in the brains of elderly people. Motor inhibition and cognate measures have shown to be related to falls in elderly people, however, so far there has been no link to a biological mechanism for this correlation. Using MRI data from The Irish Longitudinal Study on Aging (TILDA), we investigate whether the known association between motor inhibition and falls can be linked to medical imaging data.

7.2 Stopping Networks and Falling in the Elderly: Relating White Matter Architecture in the Brain's Inhibitory Control Networks to Real-World Metrics of Falling in Elderly Adults

7.2.1 Abstract

There is a complex and largely undefined relationship between cognitive decline with ageing, and the occurrence of falling. Recent research has made progress towards unpacking this association, revealing that an individual's capacity to quickly inhibit a planned or initiated action emerges as a key factor. In the current longitudinal investigation, we

measured the microstructural integrity of white matter fibre tracts connecting key nodes in the inhibitory control brain network in a large sample ($n = 414$) of older adults. We modelled their self-reported frequency of falling over a four-year period with fibre density of pathways corresponding to the direct and hyperdirect cortical-subcortical loops implicated in inhibitory control. Only connectivity between right Inferior Frontal Gyrus (rIFG) and right Subthalamic Nucleus (rSTN) was implicated in falling. This was observed cross-sectionally by modelling self-reported falling that had already occurred in the time preceding structural brain measurements. The rIFG-rSTN connectivity was not, however, predictive of future falling when measured two and four years later. Higher Apparent Fibre Density (AFD) values in the rIFG-rSTN pathways were associated with greater likelihood of falling, but only in combination with low levels of physical activity. No such relationships existed for selected control brain regions that are not implicated in inhibitory control. The direction of this effect was counterintuitive and warrants further longitudinal investigation into whether fibre density changes over time in a manner correlated with falling and mediated by physical activity.

7.2.2 Introduction

It is now well established that as higher order cognitive abilities decline with ageing, the incidence of falling increases proportionally (Amboni et al., 2013; Ambrose et al., 2013; Herman et al., 2010; Kearney et al., 2013; Li et al., 2018; Mirelman et al., 2012; Montero-Odasso et al., 2012; Muir et al., 2012). The structural and functional neural mechanisms underlying this relationship remain, however, undefined. In-depth behavioural testing has revealed that one specific facet of executive function, inhibitory control, is especially predictive of falling. In a longitudinal study, Mirelman et al., (2012) demonstrated that an individual's capacity for effective inhibitory control measured by computerised tests was predictive of fall prevalence in the subsequent 5-year period. This suggests that response inhibition, the ability to suppress highly automatic action in situations where such instinctive action is unwarranted (Fuster, 2008), may play a significant role in fall prevention. Furthermore, response inhibition is closely related to cognitive flexibility or the ability to adapt to complex and rapidly changing environments (Diamond, 2013) which correlates with fall prevalence (Kearney et al., 2013; Pieruccini-Faria et al., 2019). While the ability to stop may seem an unusual prerequisite for effective balance control, we often need to rapidly adapt our posture while navigating real-world settings. This entails occasional but appropriate suppression and revision of reflexive movements.

Beyond correlational data linking cognitive performance with falls, there have been several laboratory-based studies showing empirically how response inhibition contributes to postural equilibrium (Cohen et al., 2011; England et al., 2021; Potocanac et al., 2014; Rydalch et al., 2019; Sparto et al., 2012). The aforementioned studies focused on the execution of rapid stepping, since change-of-support reactions are often needed to regain balance (Maki & McIlroy, 1997). Older adults make more anticipatory postural adjustment errors during a choice reaction voluntary step task compared with younger adults (Cohen et al., 2011). In this case, initial acceptance of body load onto the wrong stance leg needed to first be corrected before shifting weight onto the other leg to allow the step to proceed. This led to increased choice-reaction times. Interestingly, the same study also revealed that Stroop task performance correlated with anticipatory postural adjustment errors preceding the step. The authors surmised that what may underlie an increased choice reaction time for older adults could in fact be a deficit in response inhibition versus a generic drop in processing speed due to age. Accordingly, Schoene et al. (2017) revealed that inhibitory choice reactive stepping time was associated with falls independently of reduced processing speed, lack of attention, or balance impairment. See Rey-Mermet et al. (2018), Rey-Mermet & Gade (2018), and Verhaeghen (2011) for a more nuanced discussion on the topic of inhibitory deficits and ageing.

We have recently demonstrated that performance on a balance recovery step task was correlated with speed of response inhibition in a computerised test of inhibitory control (England et al., 2021; Rydalch et al., 2019). These results, holding true for both young and older adults, suggest a common neural mechanism underlying inhibitory performance on a seated task with finger responses and a whole-body postural response to regain balance.

The underlying mechanisms of response inhibition (Enz et al., 2021; Jana et al., 2020) has received much attention in the field of cognitive psychology in a wide range of disorders (Penadés et al., 2007; Slaats-Willemse et al., 2003; Whelan et al., 2012). Using neuroimaging three underlying neural networks of response inhibition have been identified: the right inferior frontal cortex (rIFC), the presupplementary motor area (preSMA), and the subthalamic nucleus (STN) (Aron et al., 2007; Aron & Poldrack, 2006; Swann et al., 2012). Coxon et al. (2012) demonstrated that these nodes, and the strength of connectivity between them, are related to performance on response inhibition tasks. They showed that the integrity of white matter connections between the rIFC and the STN predicted response inhibition task performance and so did tract strength between preSMA and STN, but only in older adults.

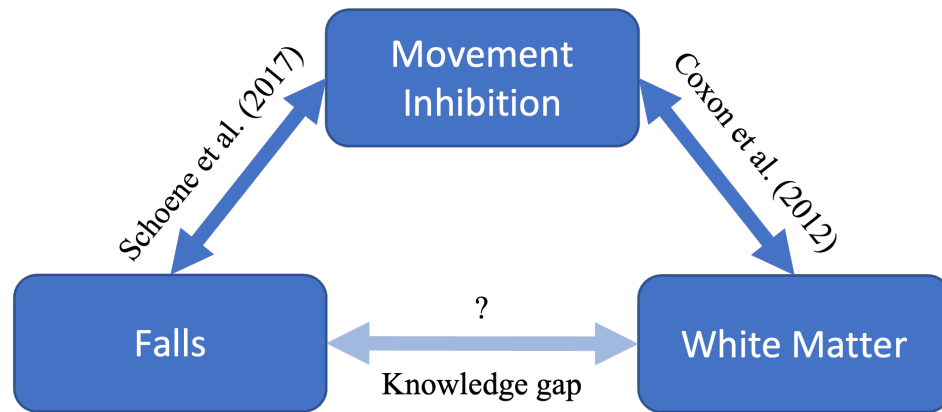


Figure 7.1. Research Gap. Theoretical framework. Schoene et al. (2017) have shown that improved performance on movement inhibition tasks is associated with a reduced number of falls in the real world. Coxon et al (2012) have shown that improved performance on movement inhibition tasks is associated with higher fractional anisotropy (FA) in right IFC and stronger connectivity between left preSMA and left STN, only in elderly people). We therefore wish to test whether individuals who fall less may show stronger white matter microstructure in the regions identified as key nodes for inhibitory control.

The theoretical framework has been outlined in figure 7.1. We hypothesise that there will be an association between white matter structures related to the motor inhibition network (STN, preSMA, and right Inferior Frontal Gyrus - rIFG) and real world falls.

The present study makes use of an extensive data set from the Irish Longitudinal Study on Aging (TILDA), which is a large-scale, longitudinal study with data on cognitive function, socioeconomic status, education, health history and many other variables to provide insight into the aging process from a broad perspective. Brain scans were collected from a subgroup (n=519) of TILDA participants. These scans were used to analyse white matter microstructural integrity between established nodes in the neural stopping network and determine if this was related to self-reported falls. We predicted that individuals with diminished connectivity between these specific networks would be more likely to experience falls. Overall, this study aims to provide insight into the neural mechanism underlying a specific cognitive ability - inhibitory control - and its relationship with fall prevalence in older adults.

7.2.3 Methods

Participant Recruitment

TILDA is a prospective, longitudinal cohort study that collects health, economic and social data from a nationally representative sample of community-dwelling Irish residents

aged 50 and over (Kearney et al., 2013). Ethical approval for the TILDA study was obtained from the Faculty of Health Sciences Research Ethics Committee the Trinity College Dublin Research Ethics Committee. Signed informed consent was obtained from all respondents prior to participation. Additional ethics approval was received for the Magnetic Resonance Imaging (MRI) sub-study from the St James's Hospital/Adelaide and Meath Hospital, Inc. National Children's Hospital, Tallaght (SJH/AMNCH) Research Ethic Committee, Dublin, Ireland. Those attending for MRI were also required to complete an additional MRI-specific consent form.

We analysed participant data collected at waves 3, 4 and 5 of the study. The data collection waves are approximately two years apart. Wave 1 was collected in 2009-2010, wave 2 was collected in 2012, wave 3 was collected in 2014-2015, wave 4 was collected in 2016, and wave 5 was collected in 2018. A collection for wave 6 is currently ongoing.

Neuroimaging data was collected at wave 3 (Whelan & Savva, 2013). Of all participants attending the wave 3 health assessment centre, a random subset was invited to return for multi-parametric brain MRI at the National Centre for Advanced Medical Imaging (CAMI) at St James's Hospital, Dublin. Participants with mild cognitive impairment and stroke may exhibit different fall profiles to those noted for typically ageing individuals and introduce additional heterogeneity (Campbell & Matthews, 2010; Härlein et al., 2009; Lamb et al., 2003; Sheridan & Hausdorff, 2007; Simpson et al., 2011). Therefore, we excluded participants with Montreal Cognitive Assessment (MoCA) (< 20) or Mini Mental State Examination (MMSE) (< 24) scores at wave 3, and additionally individuals with history of stroke or occurrence of stroke between data collection waves in the analysis.

Demographic variables applied as control variables in the models are presented in Table 7.1. They include age, sex, medical history as well as education levels, physical impairment, blood pressure, and polypharmacy (See Donoghue et al., 2018, for an overview of the TILDA Cohort).

MRI Protocol

Participants were briefed on the MRI protocol ahead of acquisition, which comprised a variety of scans including structural T1 weighted images and Diffusion Weighted Imaging (DWI) sequences. Scans were acquired via 3T Philips Achieva system and 32-channel head coil.

For the T1 3D Magnetisation-prepared Rapid Gradient Echo (MP-RAGE) sequence the acquisition parameters were: FOV (mm): 240 x 240 x 162; voxel size (mm): $0.8 \times 0.8 \times 0.9$; SENSE factor: 2; TR: 6.7 ms; TE: 3.1 ms; flip angle: 8° ; acquisition time 5:24 minutes.

DWI images were acquired with 66 slices in transverse plane with field of view 244 x 244 x 140mm; voxel size (mm): $1.9 \times 1.9 \times 2.0$; SENSE factor 2; TR: 12887 ms; TE: 55 ms; flip angle: 90° ; Diffusion was measured along 61 noncollinear directions ($b = 1200$ s/mm²) preceded by a nondiffusion - weighted volume (reference volume, $b = 0$ s/mm²). Total DWI acquisition time was 17:31 minutes.

DWI Preprocessing

DWI data were processed using ExploreDTI (Leemans et al., 2009). Images were corrected for subject motion and eddy currents using the procedure described in Leemans and Jones (2009). Tensor estimation was performed using the iteratively reweighted linear least-squares approach (Veraart et al., 2013). Fibre trajectories were computed with Constrained Spherical Deconvolution (CSD) based tractography (Tournier et al., 2007) using recursive calibration of the response function to optimise the estimation of the Fibre Orientation Distribution (FOD) functions (Tax et al., 2014). A uniform grid of tractography seed points at a resolution of $2 \times 2 \times 2$ mm³ was used with an angle threshold of 30 degrees, an FOD threshold of 0.1, and maximum harmonic order of eight. The median number of streamlines computed for each participant was 55,221 (IQR 8665). A restricted tractography analysis was performed subsequently to reconstruct streamlines passing through pairs of Regions of Interest (ROIs) that form part of the Shen 268 atlas (Shen et al., 2013). Reconstructed fibre trajectories for each individual were quantified in terms of the (median) Fractional Anisotropy (FA), Apparent Fibre Density (AFD), Mean Diffusivity (MD), and Radial Diffusivity (RD), which are all measures that reflect the directional coherence of intracellular water diffusion. Using CSD for tractography rather than the traditional diffusion tensor model allows calculation of AFD, a measure of microstructural white matter integrity that performs better than standard FA in regions with densely crossing fibres (Dell'Acqua & Tournier, 2019). As AFD provides a superior measure, we focussed our inferential statistics on this metric, but have provided comparable results with FA in the supplementary material for completeness and to allow comparison with previous research studies.

Statistical Analysis of the Demographic Variables

Statistical analysis of the demographic variables at wave 3 were performed using independent two-sample t-tests for age, sex, impairment and number of medications, and chi-square tests for the variables education, hypertension, and physical activity.

Logistic Regression

A logistic regression model was used to investigate the association between white matter structures connecting selected ROI and whether older individuals reported falling. The model was created in RStudio (RStudio Team, 2022). For each ROI a logistic model was generated. The binary dependent variable was whether participants had a fall (1) or did not fall (0) between wave 3 (2014-2015) and wave 5 (2018). The independent variables of interest were the respective measurements of reconstructed fibre trajectories for each ROI-ROI pair. There were 6 independent control variables: age, sex, education, number of medications (polypharmacy), blood pressure, and a measure of physical impairment.

Regions of Interest

The Shen 268 atlas was used (Figure 7.2 A-D), which is a parcellation of the brain into 268 areas based on resting functional state data (Shen et al., 2013). We selected five ROIs representing the movement inhibition network: the right Inferior Frontal Gyrus (rIFG), the left and right Subthalamic Nuclei (r/l STN), and the left and right presupplementary Motor Area (r/l preSMA, see Figure 7.2 E-I). The tractography was conducted in a hypothesis driven manner between restricted pairs of nodes based upon structural networks known to mediate inhibitory control (Table 7.2). To allow for comparisons between ROIs, the AFD values were z-transformed.

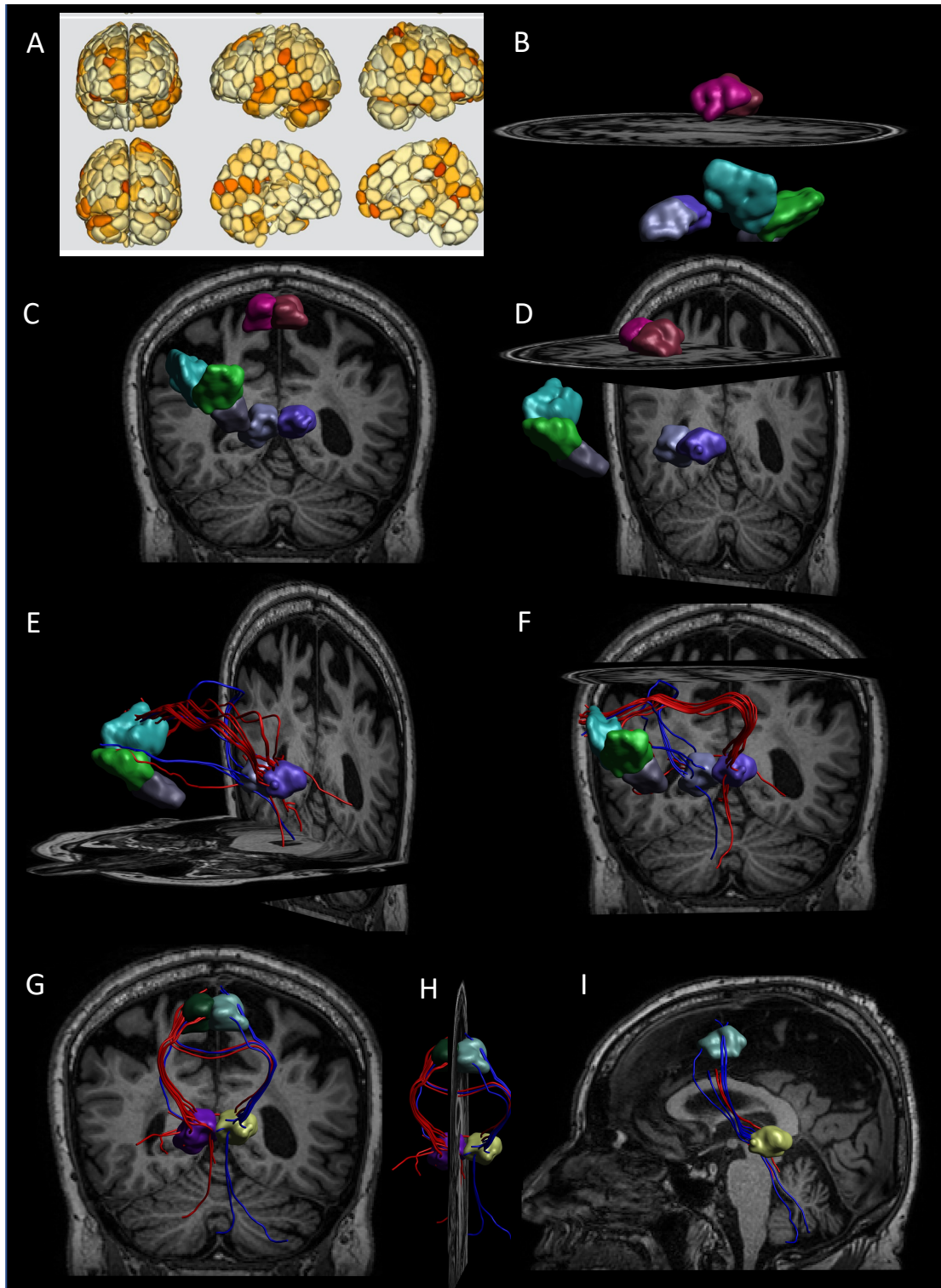


Figure 7.2. Regions of Interest and Reconstructed Streamlines. Panel A shows the Shen atlas parcellation that was used, with ROIs shown in Panels C-D selected for analysis. Panels E-F show different viewpoints of the ROIs with reconstructed streamlines passing between right and left STN and right IFG for one representative participant. Panels G,H and I show different viewpoints of reconstructed streamlines passing between bilateral STN and preSMA.

Control Analysis

As an additional experimental control, a separate tractography analysis between selected control regions and the r/l STN was performed. As a control region for the rIFG, the left IFG was chosen for topographical similarity but different functionality (Amunts & Zilles, 2012; Aron et al., 2014; Deng et al., 2017; Du et al., 2020). For the right/left preSMA control region, we chose the right/left Fusiform Face Area (FFA) as a pair of symmetrical areas not related to movement inhibition (Burns et al., 2019). Table 10.14 in the supplementary material describes the ROI characteristics.

Control variables

We included 6 control variables known to influence fall rates in our original logistic regression model: age, sex, education, blood pressure, impairment score and polypharmacy. An additional variable of physical activity was added for the post-hoc analysis of the results. Age is known to increase fall rate and was left untransformed as a numerical value (Chang et al., 2015; Deandrea et al., 2010; Franse et al., 2017; Karlsson et al., 2013). Female sex increases the severity of falls due to more prevalent osteoporosis and may increase fallrate, although the findings on the latter are inconsistent (Deandrea et al., 2010; Franse et al., 2017; Karlsson et al., 2013). Education is known to correlate with a wide array of neurologically relevant characteristics. Education serves as an indirect measurement of socioeconomic status aside from its protective effects against neurodegeneration. Both socioeconomic status and neurodegenerative processes have been discussed in their relation to falls in the older population (Khalatbari-Soltani et al., 2021; Then et al., 2016). In the linear models education was coded as a numeric variable with numbers 1-3 for primary, secondary and tertiary education.

Blood Pressure (BP) measurements were categorised after clinical criteria. For Systolic BP the four thresholds were: Normal <120, Elevated < 130, Hypertension 1 < 140, Hypertension 2 > 140, and for Diastolic the four thresholds were: Normal <80, Elevated < 80, Hypertension 1 < 89, Hypertension 2 > 90. If an individual presented two different categorisations for systolic and diastolic blood pressure, the higher BP category was chosen. High blood pressure may protect against falls caused by syncope due to low blood pressure (Butt et al., 2012), however, contradictory results exist (Ha et al., 2021).

Physical Impairments are known to increase fall rates. Our impairment score, recorded in TILDA as a series of questions asking how difficult activities of daily life were, was summed for each participant resulting in a score of 1-12 (Deandrea et al., 2010; Ha et al., 2021). Different types of drugs, such as antihypertensives, antiepileptics, sedatives and

psychotropics are known to affect fall rate. Therefore, the number of medicines used by a participant was included in the model as measure of medicinal drug use (Bloch et al., 2011; Deandrea et al., 2010; Hartikainen et al., 2007).

For an additional analysis the variable of physical activity was used. Physical activity was coded per the International Physical Activity Questionnaire (IPAQ) standard (Craig et al., 2003). The IPAQ asks participants to note the amount of time they spent doing vigorous, moderate or walking activities and gives them different weights to calculate a score and categorise participants into high, moderate and low physical activity.

Predictive Model

For the logistic model aiming to predict future falling, fallers at wave 3 were removed, and fallers at waves 4 and 5 were aggregated and labelled “fallers after wave 3”. Other parameters were the same as for the cross-sectional model.

7.2.4 Results

Demographics

Fallers had a significantly higher number of impairments $t(412) = 2.3738$, $p = 0.018$, and a significant difference in the proportion of blood pressure categories between groups, $X^2(3) = 17.0452$, $p = 0.00069$. The significant result for blood pressure is driven by hypertension 1. Without hypertension 1 the result loses significance $X^2(2) = 0.25$, $p = 0.88$.

Prevalence of Falling

For the cross-sectional analysis our criteria resulted in the inclusion of 414 participants that underwent MRI acquisition at wave 3. Ninety seven of the 414 participants at wave 3 reported having fallen since the last interview. For the predictive analysis our criteria resulted in the inclusion of 317 participants of which 96 fell between waves 3 and 4, or between waves 4 and 5.

Table 7.1*Basic Demographic Variables of Participants at Wave 3*

Selected Demographic Variables			Fallers W3 (n = 97)	Nonfallers W3 (n = 317)	p-Value
Age		Mean (sd)	69.21 (8.23)	68.2 (7.39)	0.2524 ^b
Sex	Male	n (%) ^a	42 (43.3)	158 (49.8)	0.31 ^c
	Female	n (%) ^a	55 (56.7)	159 (50.2)	
Education					0.29 ^c
	Level 1	n (%) ^a	22 (22.7)	51 (16.1)	
	Level 2	n (%) ^a	36 (37.1)	119 (37.5)	
	Level 3	n (%) ^a	39 (40.2)	147 (46.4)	
Impairment		Mean (sd)	2.05 (1.88)	1.58 (1.65)	0.018^b
Blood Pressure					0.00069^c
	Normal	n (%) ^a	25 (25.8)	69 (21.8)	
	Elevated	n (%) ^a	16 (16.5)	39 (12.3)	
	Hypertension 1	n (%) ^a	10 (10.3)	99 (31.2)	
	Hypertension 2	n (%) ^a	46 (47.4)	110 (34.7)	
Number of Meds		Mean (sd)	2.94 (2.34)	2.5 (2.52)	0.13 ^b
Physical Activity		n	92	302	0.21 ^c
	Low	n (%) ^a	35 (38.0)	106 (35.1)	
	Moderate	n (%) ^a	40 (43.5)	113 (37.4)	
	High	n (%) ^a	17 (18.5)	83 (27.5)	

Note: Participants were grouped into fallers and nonfallers. ^a Valid percent ^b Independent two-sample t-test ^c Chi-square test over all levels and categories.

Associative (Cross-Sectional) Logistic Regression Results

The results of the cross-sectional logistic regression are depicted in Table 7.2. As the analysis tested six main hypotheses, the significance threshold was adapted using a Bonferroni correction for six tests yielding a new critical alpha of 0.0083. The model using the AFD values between the rIFG and rSTN achieved a p of 0.005. This implies that an increase in AFD of 1 standard deviation in the tracts connecting rIFG and rSTN significantly increased the odds of falling by 1.49 (CI: 1.13, 1.98).

Table 7.2

Association Between Microstructural Integrity in Inhibitory Control Networks and Odds of Falls in the Elderly

Region	Odds	CI Low	CI High	P Value	Chi Square Fit	n
r IFG - r STN	1.49	1.13	1.98	0.005	0.00003	360
l preSMA - r STN	1.38	0.93	2.05	0.113	0.00046	200
r IFG - l STN	1.28	0.91	1.8	0.161	0.00172	237
r preSMA - r STN	0.84	0.61	1.16	0.288	0.02334	257
r preSMA - l STN	0.91	0.65	1.26	0.557	0.00976	228
l preSMA - l STN	0.97	0.73	1.29	0.855	0.00096	343

Note: Results of a logistic regression showing the association between the tractography of ROIs and the risk of falling in older people. A single result (rIFG to rSTN) is significant after correcting for multiple comparisons (Bonferroni, new pthreshold: *0.0083*), in bold. The Chi Square and number of observations in the model are included.

The model fulfilled all assumptions for a logistic regression (see supplementary material 5.2). A chi square fit showed that the model was a good fit for the data ($\chi^2 = 0.00003$), and the McFadden R^2 Improved from 0.054 (model without AFD) to 0.094 by including the variable of interest. An additional observation is that the control variable blood pressure with category hypertension 1 significantly decreased the odds of falling by 0.18 (CI: 0.065, 0.48, p-value: 0.00067) (Figure 7.3).

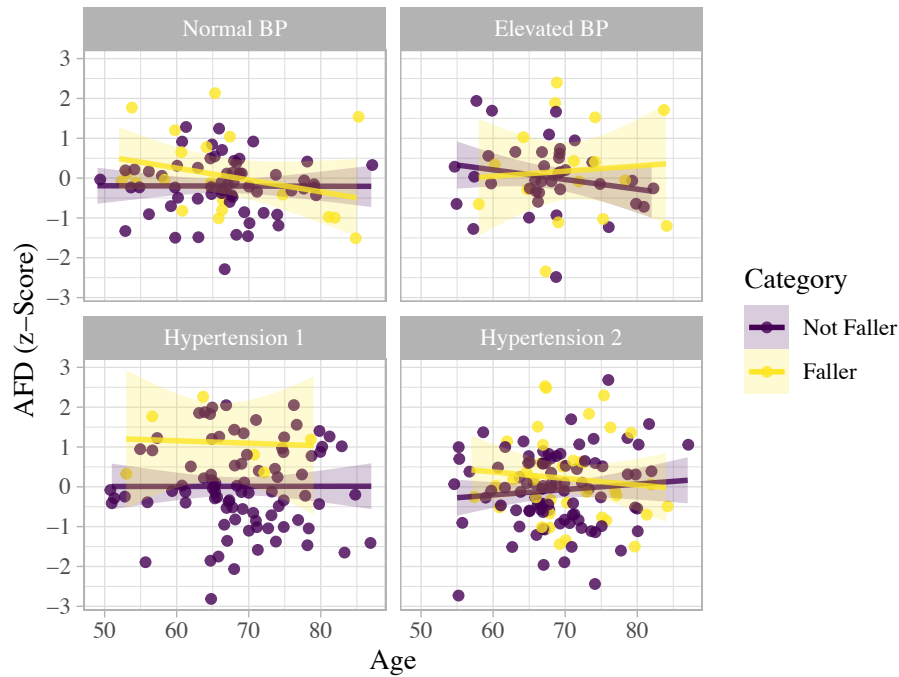


Figure 7.3 AFD Scores Split by BP. This Figure shows rIFG – rSTN fibre density (AFD) by Age, separated by Blood Pressure (BP) categories, with separate lines for fallers and nonfallers. There were significantly less fallers in the ‘Hypertension 1’ category. In the cohort with normal blood pressure, there are a total of 84 participants, 21 (25%) of which fell. For elevated blood pressure there are a total of 54 participants, 16 (29.63%) of which fell. For hypertension 1 there are a total of 93 participants, 6 (6.45%) of which fell. For hypertension 2 there are a total of 129 participants, 37 (28.68%) of which fell.

Figure 7.4 A shows that older individuals who fell ($M = 0.23$, $SD = 1.1$) had higher AFD values in the white matter pathways connecting rIFG to rSTN (Nonfallers: $M = -0.066$, $SD = 0.92$, directional Mann Whitney U test, $W = 9716$, $p = 0.035$). This effect was most pronounced in the 50-65yr old fallers, as AFD values appeared to be lower in the older 65+ fallers. Nonfallers show no such trend in AFD values cross sectionally over the age range. We investigated this post hoc by adding an age-AFD interaction term to the rIFG-rSTN model. The age-AFD interaction term did not reach significance, reducing odds of a fall by 0.038 (CI: 0.072, 1.28, $p = 0.058$) while the AFD term was still significant, increasing the odds of a fall by 21.87 (CI: 16.47, 29.04, $p = 0.03$). The increase in odds for the rIFG to rSTN AFD value is mathematically inflated in the model with the interaction term, as the value features twice in the model as part of the interaction and main effect. It is also further inflated due to the comparatively high numeric range of age.

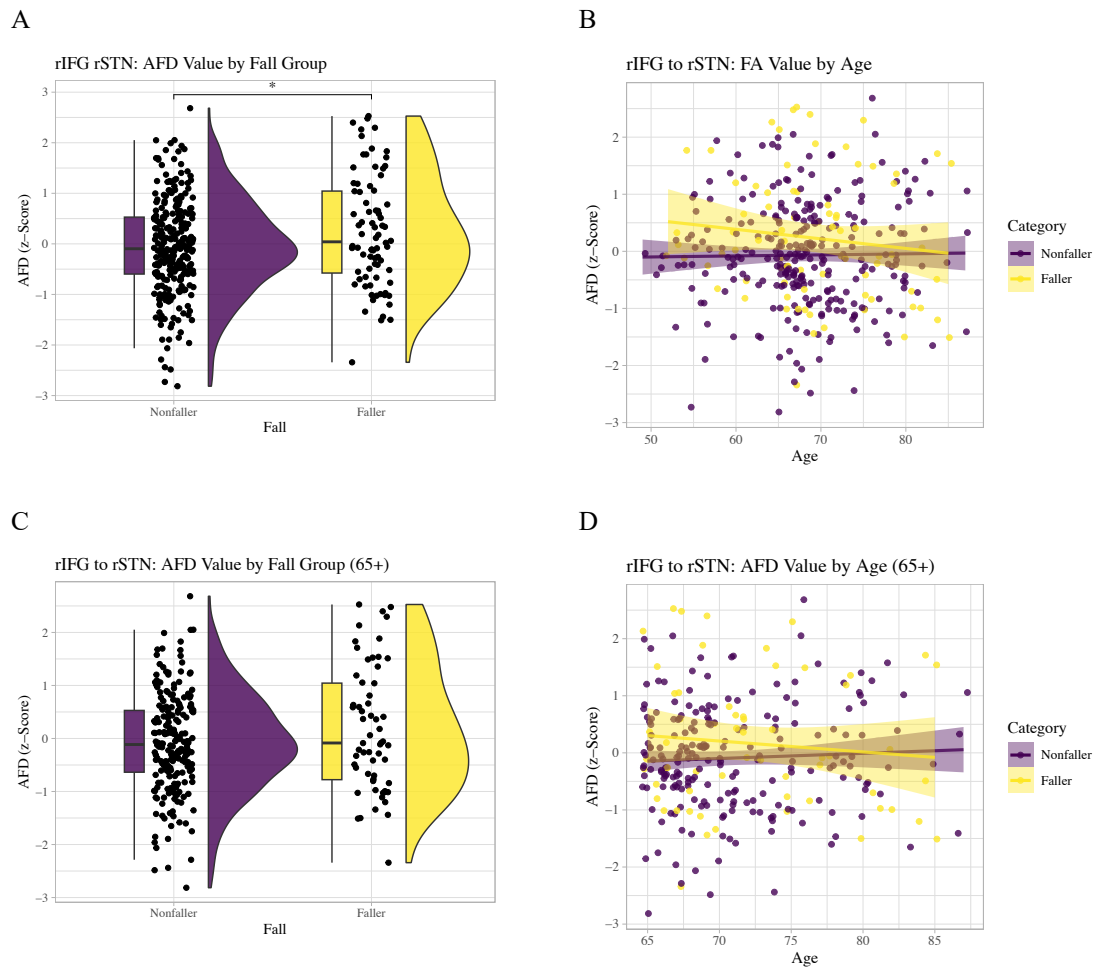


Figure 7.4. AFD Values by Group and Age. Figure A shows differences in the distribution of fallers vs nonfallers. Fallers seem to have overall higher AFD values. Figure B shows the difference in distribution according to age. Fallers also have a higher average AFD value, although this relationship is dependent on age. Figure C and D show the difference in distribution of fallers, but only including fallers of age 65 or more.

Constraining the analysis to individuals aged 65+ has no effect on the overall distribution (Figure 7.4 C & D). However, no significant results were found using a sample of people aged 65 or more – likely due to the reduced sample size.

Interaction with Physical Activity

We hypothesized that higher AFD values indicative of dense white matter connectivity in older people would be associated with lower risk of falling. However, this relationship was not found in our data – instead, we found that higher AFD led to increased fall risk. We investigated this relationship deeper, hypothesizing that more active older people may be generally healthier and have higher AFD values, and be more likely to fall due to greater physical activity than their sedentary counterparts.

Table 7.3

Association Between Microstructural Integrity in Inhibitory Control Networks and Odds of Falls in Older Adults

Region	Odds	CI Low	CI High	p-Value	Chi Square Fit	n
rIFG - r STN	2.31	1.42	3.78	0.00082	0.000061	360
l preSMA - r STN	1.31	0.74	2.3	0.35	0.0098	200
l preSMA - l STN	1.23	0.76	2	0.41	0.0037	343
r preSMA - l STN	0.8	0.43	1.49	0.48	0.041	228
r preSMA - r STN	0.83	0.47	1.47	0.53	0.1	257
rIFG - l STN	1.06	0.61	1.86	0.83	0.0083	237

Note: Results of a logistic regression showing the association between the tractography of ROIs and the risk of falling in older people when accounting for physical activity. A result (rIFG to rSTN) is significant after correcting for multiple comparisons (Bonferroni, new pthreshold: 0.0083), the pis bold. The Chi Square and number of observations of the model are included.

Adding an interaction between physical activity level and AFD values to the model required the inclusion of a main effect term. Therefore, the updated logistic regression model contained two new elements: A term for physical activity and the term for the interaction between physical activity and AFD values.

The results of the cross-sectional logistic regression are depicted in Table 7.3. As the analysis contains six hypotheses, the significance threshold was adapted using a Bonferroni correction for six tests yielding an adjusted alpha of 0.0083. The model using the AFD values between the rIFG and rSTN achieved a p value of 0.00082. This means that an increase in rIFG - rSTN AFD by 1 standard deviation significantly increased the odds of falling by 2.31 (CI: 1.42, 3.78). The McFadden Pseudo R squared of this model improves to 0.12 compared to a model with no AFD and no AFD * physical activity interaction.

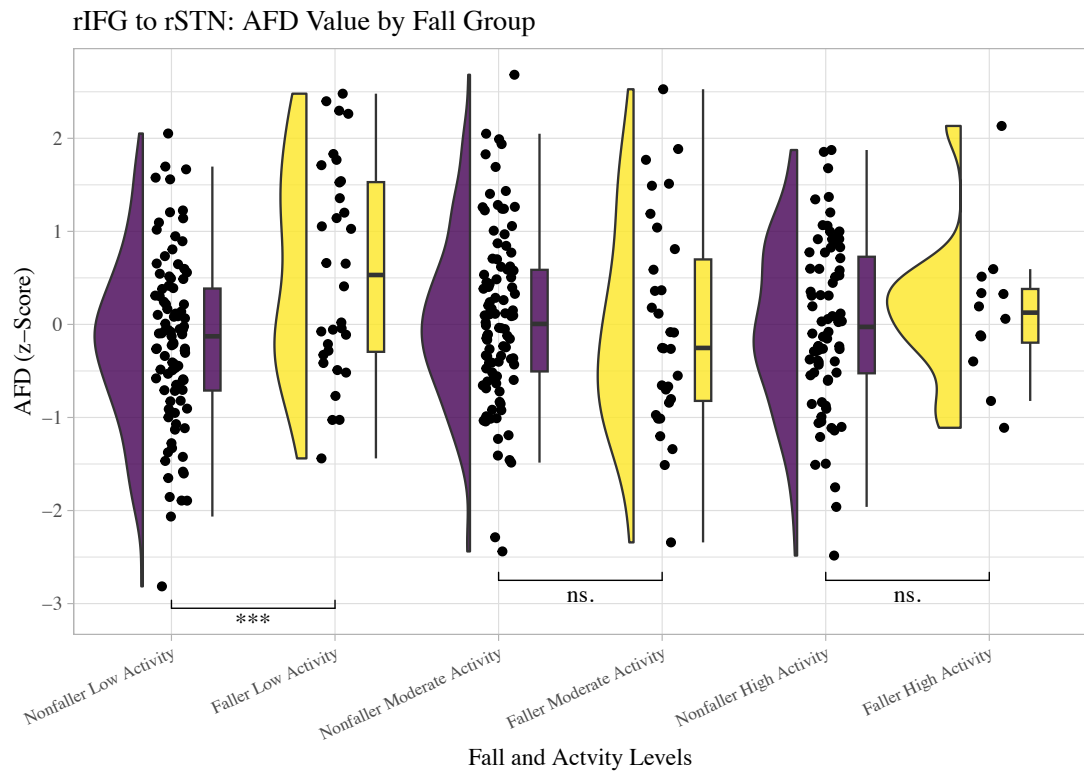
Moderate physical activity increased the odds of falling by 1.44 (CI: 0.75, 2.75), although not significantly ($p = 0.28$). However, the interaction term of moderate physical activity and AFD value significantly ($p = 0.014$) decreased the odds of falling by 0.44 (CI: 0.22, 0.84). High physical activity does not significantly affect outcomes, neither as a main or interaction effect.

In figure 7.5 it is visible that AFD is significantly ($t(125) = -3.87$, $p = 0.00018$) higher for fallers ($M = 0.58$, $SD = 1.12$; nonfallers: $M = -0.19$, $SD = 0.92$) that are not physically active, however, the same is not true for moderately active (Fallers: $M = -0.023$, $SD = 1.13$; nonfallers: $M = 0.066$, $SD = 0.92$; $t(127) = 0.44$, $p = 0.66$), or highly active older people (Fallers: $M = 0.13$, $SD = 0.82$; nonfallers: $M = -0.007$, $SD = 0.9$; $t(86) = -0.5$, $p = 0.62$). The

data presented in figure 7.5 B further suggests that much of the interaction between falling and age is cleared up when the model accounts for physical activity.

Figure 4

A



B

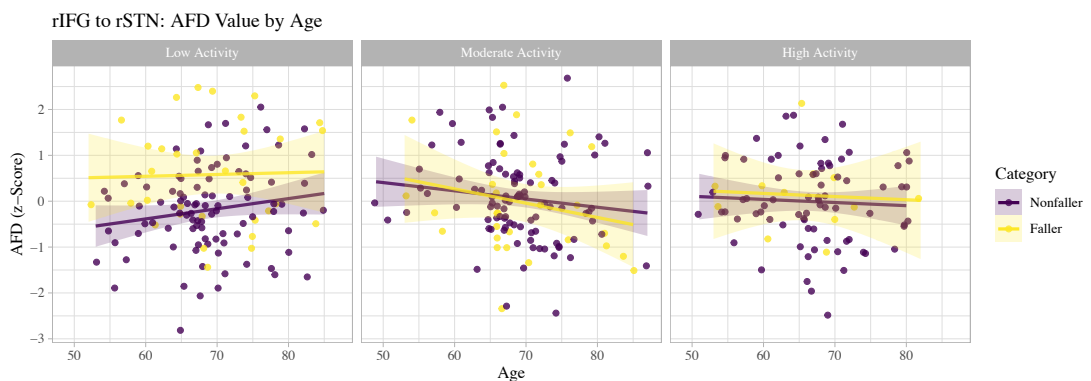


Figure 7.5. rIFG to rSTN AFD Values by Group and Age. Figure A shows differences in the distribution of fallers vs nonfallers. Fallers seem to have overall higher AFD values in the low activity condition, but not in high or moderate physical activity. Figure B shows the difference in distribution according to age. Fallers also have a higher average AFD value, and this relationship is less dependent on age when accounting for physical activity.

ROI Subregion Analysis

Table 7.4

Association Between Microstructural Integrity in Inhibitory Control Networks and Odds of Falls in Older Adults

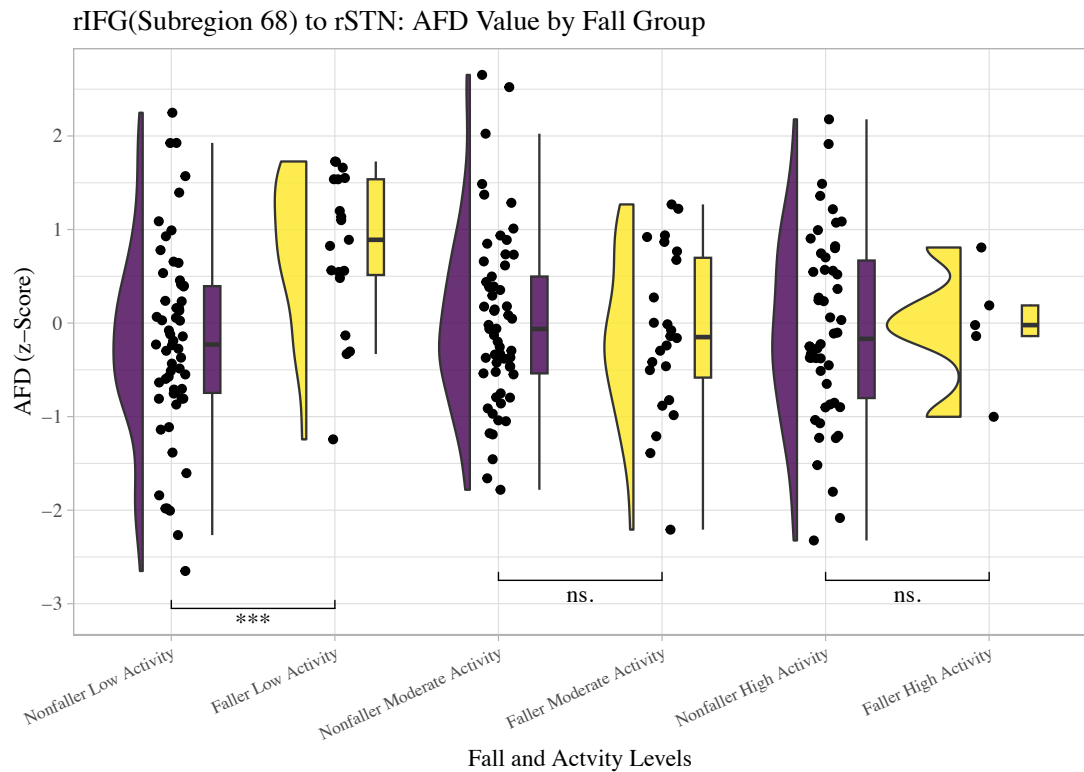
Region	Odds	CI Low	CI High	p-Value	Chi Square Fit	n
rIFG (subregion 68) - r STN	3.59	1.7	7.56	0.00079	0.000027	221
rIFG (subregion 79) - r STN	2.08	1.06	4.05	0.032	0.007	249
rIFG (subregion 79) - l STN	1.79	0.84	3.8	0.13	0.16	130
rIFG (subregion 90) - r STN	0.58	0.28	1.19	0.14	0.21	186
rIFG (subregion 68) - l STN	1.22	0.45	3.31	0.7	0.0016	139
rIFG (subregion 90) - l STN	8.93E+21	0	-	1	0.0014	35

Table 4. Results of a logistic regression showing the association between the tractography of ROIs and the risk of falling in older people when accounting for physical activity. One result (rIFG to rSTN) is significant after correcting for multiple comparisons (Bonferroni, new pthreshold: 0.0083), the p is bold. The Chi Square and number of observations of the model are included.

The results of the cross-sectional logistic regression testing three further subdivisions of rIFG are depicted in Table 7.4. As the analysis contains six main hypotheses, the significance threshold was adapted using a Bonferroni correction for six tests yielding a new alpha of 0.0083. When analysing subregions of the rIFG, one region is significant. Area 68 - R.BA.37.10 in the Shen atlas – near to the parahippocampal gyrus is significant ($p=0.00079$). Increases of 1 SD of AFD in this region increases the odds of falling by 3.59 (CI: 1.7, 7.56).

Compared to the model using the whole rIFG structure, the McFadden pseudo R squared improves from 0.054 in a model with no AFD or AFD and physical activity interaction term to 0.2.

A



B

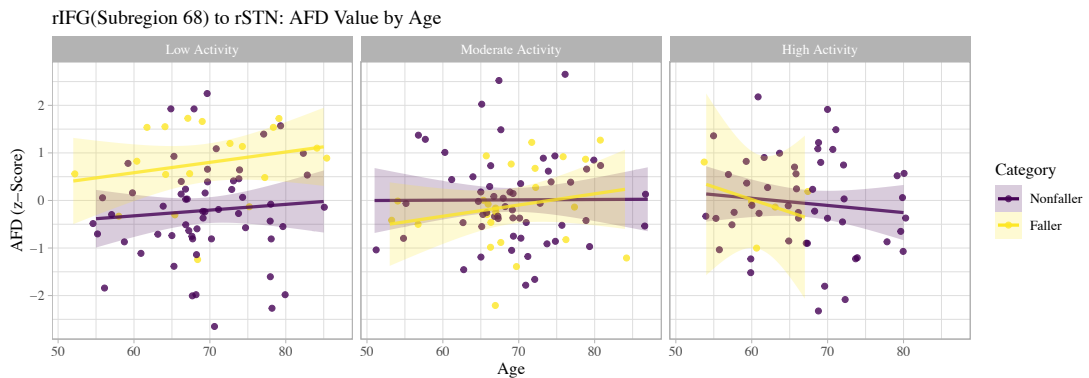


Figure 7.6. rIFG (Subregion) to rSTN AFD Values by Group and Age. Figure A shows differences in the distribution of fallers vs non fallers. Fallers seem to have overall higher AFD values in the low activity condition, but not in high or moderate physical activity. Figure B shows the difference in distribution according to age. Fallers also have a higher average AFD value.

In this model, moderate physical activity significantly increased the odds of falling by 2.6 (CI: 1.05, 6.4, $p = 0.038$). Similarly, the interaction term of moderate physical activity and AFD value significantly ($p = 0.006$) decreased the odds of falling by 0.27 (CI: 0.1, 0.68). High physical activity did not significantly affect outcomes, neither as a main or interaction effect.

Looking at figure 7.6 we can see that AFD is significantly ($t(76) = -3.85, p = 0.00025$) higher for fallers ($M = 0.79, SD = 0.83$; nonfallers: $M = -0.21, SD = 1.03$) that are not physically active, however, the same is not true for moderately active (Fallers: $M = -0.12, SD = 0.88$; nonfallers: $M = 0.012, SD = 0.94$; $t(79) = 0.59, p = 0.56$), or highly active older people (Fallers: $M = -0.033, SD = 0.65$; nonfallers: $M = 0.06, SD = 1$; $t(53) = -0.058, p = 0.95$).

Predicting Future Falling from Structural Brain Data

We combined data on falling that occurred at any point following the MRI scan at wave 3 until wave 5. The results of the predictive logistic regression are depicted in table 7.5. As the analysis contains six main hypotheses, the significance threshold was adapted using a Bonferroni correction for six tests yielding a critical alpha of 0.0083. AFD of white matter pathways connecting any of the aforementioned ROIs did not predict future falling at waves 4 or 5. (Table 7.5).

Table 7.5

Prediction of Fall Risk in Older Adults by White Matter Microstructure

Region	Odds	CI Low	CI High	P Value	Chi Square Fit	n
l preSMA - l STN	1.42	1.05	1.92	0.023	0.023	265
r preSMA - r STN	0.74	0.52	1.06	0.097	0.017	200
r preSMA - l STN	0.81	0.54	1.2	0.29	0.026	171
r IFG - r STN	1.04	0.78	1.39	0.776	0.125	276
r IFG - l STN	0.97	0.66	1.42	0.861	0.330	180
l preSMA - r STN	0.98	0.69	1.4	0.926	0.387	156

Note: Results of a logistic regression showing the prediction of the risk of falling in older adults using Apparent Fibre Density in pathways connecting targeted ROIs. No result is significant after correcting for multiple comparisons (Bonferroni, new p-threshold: *0.0083*). The Chi Square and number of observations of the model are included.

The results of the predictive logistic regression accounting for physical activity are depicted in table 7.6. Again, the Bonferroni adjusted critical alpha was 0.0083. No significant association between the independent and dependent variables were observed (Table 7.5).

Table 7.6

Association Between Microstructural Integrity in Inhibitory Control Networks and Odds of Falls in Older Adults

Region	Odds	CI Low	CI High	p-Value	Chi Square Fit	n
r preSMA - r STN	0.43	0.22	0.84	0.014	0.039	201
l preSMA - l STN	1.55	0.93	2.58	0.09	0.14	268
r IFG - r STN	1.16	0.74	1.84	0.52	0.28	282
r IFG - l STN	0.92	0.5	1.7	0.79	0.34	186
l preSMA - r STN	1.04	0.64	1.69	0.88	0.24	158
r preSMA - l STN	1.03	0.5	2.12	0.94	0.12	173

Note: Results of a logistic regression showing the prediction of the risk of falling in older adults using the Apparent Fibre Density. No result is significant after correcting for multiple comparisons (Bonferroni, new pthreshold: 0.0083). The Chi Square and number of observation of the model are included.

Control ROI Analysis

To further guard against false positives, we also performed a control analysis using areas not directly implicated in inhibitory control. To maintain similarity with the experimental analyses, we still targeted bilateral STN, but instead of analysing the rIFG and preSMA connections to STN, we chose the left IFG and the FFA, areas generally not considered to be substantial components of the inhibitory control network. This yielded no significant results when any of the aforementioned models were conducted with the control regions.

Table 7.7

Association Between Microstructural Integrity in Inhibitory Control Networks and Odds of Falls in Older Adults

Region	Odds	CI Low	CI High	p-Value	Chi Square Fit	n
r FFA - r STN	1.56	1.04	2.33	0.03	0.0227	178
l IFG - l STN	0.73	0.53	1	0.05	0.0001	282
l IFG - r STN	0.84	0.59	1.21	0.36	0.0042	197
l FFA - l STN	0.85	0.54	1.35	0.5	0.0050	124
r FFA - l STN	1.34	0.44	4.08	0.61	0.2286	39
l FFA - r STN	0.92	0.42	2	0.83	0.0192	70

Note: Results of a logistic regression showing the prediction of the risk of falling in older adults using the Apparent Fibre Density. No result is significant after correcting for multiple comparisons (Bonferroni, new pthreshold: 0.0083). The Chi Square and number of observations of the model are included.

Voxel count (with rIFG and lIFG split up into their individual ROIs) between control ROIs ($M = 3391$, $SD = 738.99$) and experimental ROIs ($M = 3919.86$, $SD = 899.14$) did not differ significantly ($t(11.567) = 1.20$, $p = .25$).

Cross-Sectional Models for Control ROIs

The results of the cross-sectional logistic regression are depicted in table 7.7. As the analysis contains six main hypotheses, the significance threshold was adapted using a Bonferroni correction for six tests yielding a new alpha of 0.0083. No significant association between the independent and dependent variables were observed (Table 7.7).

Predictive Models for Control ROIs

The results of the predictive logistic regression are depicted in table 7.8. As the analysis contains six main hypotheses, the significance threshold was adapted using a Bonferroni correction for six tests yielding a new alpha of 0.0083. No significant association between the independent and dependent variables were observed (Table 8).

Table 7.8

Association Between Microstructural Integrity in Inhibitory Control Networks and Odds of Falls in Older Adults

Region	Odds	CI Low	CI High	p-Value	Chi Square Fit	n
l IFG - l STN	1.38	1	1.91	0.05	0.0015	223
r FFA - l STN	0.34	0.09	1.3	0.12	0.1150	29
r FFA - r STN	1.37	0.89	2.09	0.15	0.0081	138
l FFA - r STN	1.42	0.73	2.78	0.3	0.2500	55
l FFA - l STN	0.99	0.54	1.84	0.98	0.1232	89
l IFG - r STN	1	0.65	1.53	1	0.1557	147

Note: Results of a logistic regression showing the prediction of the risk of falling in older adults using Apparent Fibre Density. No result is significant after correcting for multiple comparisons (Bonferroni, new threshold: 0.0083). The Chi Square and number of observations of the model are included.

7.2.5 Discussion

In the current longitudinal investigation, we demonstrated a significant association between white matter fibre density in pathways connecting two key regions in the brain's inhibitory control network and falling in a large sample ($n=414$) of older participants. We tested the microstructural integrity of white matter pathways corresponding to the direct and hyperdirect cortical-subcortical loops implicated in inhibitory control and found that only connectivity between right Inferior Frontal Gyrus (rIFG) and right Subthalamic Nucleus

(rSTN) was implicated in falling. This was observed cross sectionally by modelling self-reported falling that had already occurred in the time period preceding structural brain measurements. The rIFG-rSTN connectivity was not, however, predictive of future falling when measured two and four years later. Further, no such relationships existed for selected control brain regions that are not implicated in inhibitory control. While statistically robust and surviving strict multiple comparison corrections, our key finding was counterintuitive as the direction of the effect was opposite to that which we hypothesised. Higher Apparent Fibre Density (AFD) values in the rIFG-rSTN pathways were associated with greater likelihood of falling. We performed post-hoc analyses to unravel the effect, revealing that this finding was significantly influenced by physical activity levels in the older individuals. Higher AFD values only yielded higher odds of falling in combination with low levels of physical activity. In individuals with moderate or high physical activity levels, AFD had no bearing on falling.

Having a large sample size allowed us to construct a complex logistical model with falling as the dependant variable, using a set of known influences as control variables (age, sex, education, blood pressure, polypharmacy, impairments of daily living) and the AFD values between ROIs as independent variables. We focussed our analysis on AFD instead of the traditionally reported Fractional Anisotropy (FA) values to measure white matter structures within the brain. AFD offers several advantages over FA, the most pertinent being increased accuracy for measuring crossing fibres tracts within voxels (Dell'Acqua & Tournier, 2019). The model reaffirmed the previous finding that high blood pressure may act as a protective factor against falls – likely by preventing falls due to syncope from blood pressure drops (Butt et al., 2012). A further strength of the study was that an investigation into control areas not related to movement inhibition yielded no significant results.

Coxon et al. (2012) initially established a relationship between right Inferior Frontal Cortex (rIFC) white matter structure and decreased response inhibition time in young and older adults. They additionally reported higher FA in white matter projections bilaterally between the IFC and STN in older (but not younger) adults with the fastest response inhibition times. Schoene et al. (2017) demonstrated an association between step response inhibition and real life falls and consistent with this idea, Nagamatsu et al. (2013) found hypo-activation in prefrontal brain regions during a test of inhibitory control in individuals who fell more often. Hence, we hypothesised that greater microstructural integrity of white matter pathways in these networks may predict current and future falling. While we did detect a significant relationship, our finding that the individuals with most densely connected

pathways fell more was surprising. While the study by Coxon et al (2012) used FA as the main measure of white matter microstructure, we chose to use AFD in a move towards more complex models that take into account the complexity of fibre density and directionality such as AFD. We did, however, verify that the same pattern of results reported here holds true with FA (see supplementary material for analyses). Furthermore, while FA values generally decline with increasing age, this relationship does not apply to AFD values (Choy et al., 2020). Therefore, a complex relationship between AFD and falling behaviour is likely.

As this was an observational study and the predictive models yielded no significant findings, we cannot infer causality or directionality in the relationship between fibre density and falling. The fact that individuals who fall tended to already have higher fibre density in inhibitory control pathways may be a cause or consequence of the falling. For example, it is conceivable that increased AFD values in fallers may be related to increased attention to balance and active learning processes subsequent to a fall, rather than bring pre-existing. Follow-up MRI scanning with the same cohort of participants may unpick this relationship further to disentangle whether changes in rIFG-rSTN microstructure drive changes in falling or vice versa.

To define this relationship further, we investigated the mediating effects of physical activity. By definition physical activity implies that people are engaging in behaviours that make falls more likely. It is therefore not surprising that physical activity itself leads to an increase in falling behaviour in our models. Interestingly there was no correlation between falling and AFD in those with higher physical activity levels. This warrants follow-up investigation with more objective measurement methodologies as the activity levels reported in TILDA rely on self-reported activity levels within the last 7 days of interviewing, which has been shown to be subject to over- and underestimation (Lee et al., 2011; Prince et al., 2008).

7.2.6 Conclusion

Using MRI and self-reported data from 414 participants from the Irish longitudinal study on ageing we showed that higher microstructural integrity in white matter pathways connecting the right inferior frontal gyrus and right subthalamic nucleus was associated with falling in older adults. This relationship was pre-existing at the time of structural MRI data acquisition, and therefore precludes establishing causality or directionality of the effect. Fibre density at the time of MRI data collection did not predict future falling two or four years later. Follow-up MRI data will be required in order to determine whether densely connected regions in the inhibitory control network change over time in a manner correlated

with falling, or whether this relationship is purely cross-sectional, and perhaps mediated by a third currently undefined factor.

7.2.7 Author Contribution

KR and DB conceptualised the idea for the manuscript. DB wrote the first draft of the introduction. CS completed the statistical analysis. CS wrote the first draft and edited the final draft. CS, KR, VS worked on subsequent drafts of the manuscript. KR, CS, VS DB, SK, and the TILDA Consortium provided feedback. All authors reviewed the final version and were involved in discussions throughout.

CRedit Contributions

Conceptualization: K.R. and D.B.; Data curation: S.K.; Formal analysis: C.S.; Funding acquisition: K.R.; Methodology: C.S., K.R. and D.B.; Project administration: C.S. and K.R.; Resources: S.K.; Software: C.S.; Supervision: K.R.; Visualization: C.S.; Writing – original draft: C.S.; Writing - review & editing: C.S., K.R., D.B., J.M., R.A.K., V.A.S., C.D. and S.K.

8. General Discussion

8.1 Challenges and Opportunities

In the second chapter of this thesis titled “Brain Computer Interfaces in neurorehabilitation” and its corresponding publication called “Challenges and Opportunities for the Future of Brain-Computer Interface in Neurorehabilitation” we investigated why, despite encouraging results in small scale clinical trials, poststroke rehabilitative Brain-Computer Interfaces (BCIs) are not widely used.

We found that there were many factors, most of which can be summarised as issues of scalability or justifiability. Scalability issues are caused by the cost of treatment; BCIs are expensive to buy, set up, and use, as they are complex and need trained attendees to set up and enable training. Justifiability issues are caused by the insufficient theoretical background available for BCIs; there are many different BCI designs, targeting many different BCI working mechanisms, and medical validity is spread thinly among all these designs. The heterogeneity in designs and the resulting questionable efficiency makes it difficult to justify the use of resources to acquire a BCI setup.

We suggest three different solutions that are interconnected. First, we suggest a renewed focus on the working mechanisms of BCI to increase medical validity (and therefore justifiability). Second, we suggest subdividing BCI systems to address individual BCI working mechanisms and or rehabilitative challenges. The first and second suggestion are strongly linked to increasing our understanding of how BCI may be able to help with rehabilitation, which should increase its efficiency, medical validity and homogeneity of designs. Our third suggestion is to look at contributions of nonspecific effects of BCI use such as improved cognition, to further increase medical validity.

8.2 Multiphase and Multimodal BCI

In the third chapter of this thesis titled “Transcranial Magnetic Stimulation Neurofeedback – A Multimodal, Multiphase Approach to Stroke Rehabilitation Using BCI” and the corresponding manuscript “Optimising Control of Motor Imagery-based BCIs by Priming Using TMS-Neurofeedback” we designed a two-phase, multimodal BCI system, targeting corticospinal excitability as measured with Motor Evoked Potentials (MEPs). This has been shown to be connected to motor recovery after stroke (Stinear et al., 2015) and applicable as a BCI control scheme (Ruddy et al., 2018).

The resulting two-phase BCI setup used Transcranial Magnetic Stimulation Neurofeedback (TMS-NF) to teach participants to increase or decrease their cortical excitability before moving on to a more traditional Electroencephalography (EEG) BCI with the same Motor Imagery (MI) instructions. Comparing the results of the two-phase BCI to a monophasic version using only the EEG BCI showed that healthy participants training with the two-phase BCI were faster and more accurate. Further, the patterns of their activity more closely resembled those of Motor Execution (ME) compared to participants in the monophasic group. Lastly, we found that different preprocessing of the EEG signal led to different power bands being significantly associated with BCI control, further emphasizing the need for a better understanding of how BCI control works and can help with rehabilitation.

The improved accuracy and speed of learning using the two-phase BCI is a significant achievement, as it may contribute positively towards resolving issues of motivation and fatigue – reported as one of the worst symptoms after stroke by up to 40% of survivors (Gangwani et al., 2022; Paciaroni & Acciarresi, 2019).

However, the more intriguing result is the fact that the spatial attributes of the two-phase BCI group differed from the monophasic group. This may show that including a different modality may not only improve speed and efficiency of the training time, but also change the quality of the MI. In this specific case it seems that the two-phase group developed a kind of MI that involved more posterolateral areas than the monophasic group, consistent with differences between MI and ME. This is potentially revolutionary for poststroke BCIs, as it shows the possibility to train using MI that strongly resembles ME, *before* participants regain enough dexterity to start physical rehabilitation.

However, both phases of the two-phase BCI were complex and needed trained administrators for setup and training. We therefore decided to design a simple BCI system that ideally could be used at home by stroke survivors to train without supervision. This is a continuation of the ideology presented in chapter 2; to create a BCI to solve a very specific purpose: training at home and without supervision.

8.3 The Feasibility of Multiphase and Multimodal BCI

The fourth chapter of this thesis titled “The Feasibility of Multiphase BCI Rehabilitation” and the according publication “A Wireless, Wearable Brain-Computer Interface for Neurorehabilitation at Home; A Feasibility Study”, was designed to investigate the feasibility of creating an easy to use medical BCI systems that could be used by stroke

survivors to train at home without supervision. The designed BCI used an easy to use, wireless, Bluetooth, 16 dry electrode EEG headset intended for at home training of the same MI as used by the two-phase BCI.

We found that healthy participants were able to set up, train with, and achieve control over the BCI system within six training sessions with minimal online instructions. These positive results open the way to investigate the medical benefits of a similar trial using stroke survivors. This may be an inexpensive way to increase the amount of training individual stroke survivors receive (Lang et al., 2015; Stinear et al., 2020). Further, there are nonspecific BCI effects that could improve outcomes; the ability to train at home could increase agency and motivation (Gangwani et al., 2022).

The results of this study, combined with those of the two-phase BCI in chapter three open some very interesting possibilities. It should be possible to combine both BCI to create a three-phase BCI, in which participants first train using TMS-NF, then learn to control the EEG-BCI, and then move on to train on their own at home. Together, this three-phase BCI shows considerable medical validity as in each phase a relevant BCI process is targeted (increased cortical excitability). Further, the last phase can significantly increase efficiency by allowing individuals to train without supervision. Together, these studies present a strong case for designing multimodal and multiphase BCI to increase scalability and justifiability of medical BCI.

8.4 Designing BCI for Rehabilitation

In the fifth chapter of this thesis titled “Contemporary Considerations for BCI Design in Neurorehabilitation” and the according publication “Keeping our Eyes on the Prize; Are we Losing Sight of the ‘Why’ in BCI for Neurorehabilitation?” we tried to distil the approach used in chapter three and four to designing multimodal and multiphase BCI.

We describe three main questions that should guide medical BCI design: What neural signal should be targeted? What therapeutic goal is to be achieved? What technology is appropriate to record the neural signal and achieve the therapeutic goal? We also encourage to think broadly about what signals the brain can produce, and not only about its electromagnetic properties. In traditional medical BCI design the brain is often the originator of signal but it should also be thought of as a possible processor of signal. In TMS-NF, the TMS pulse is the signal originator, and the brain is the signal processor.

While our analysis of medical BCI has been focused on poststroke rehabilitative BCI, this is not the only possible application for them. In the next chapter we talk about the possible applications of BCI for rare diseases and disorders.

8.5 BCI for Rare Diseases and Disorders

In the sixth chapter of this thesis titled “Applications of BCI for Rare Diseases and Disorders” and the according publication “Could Brain–Computer Interface Be a New Therapeutic Approach for Body Integrity Dysphoria?” we investigated how Body Integrity Dysphoria (BID) may benefit from a BCI-based approach.

We found that most therapies for BID show little to no effect leaving space for innovative treatment ideas. BID is a complex disorder with many different psychological and biological components. Therefore, it is difficult to predict the success of targeting a mechanism and incorporating it in a BCI-based therapy. However, we found multiple, possible targets for designing BCI-based therapy. We suggest that the right superior parietal lobule (McGeoch et al., 2015) may be a good target for a TMS-NF like BCI, where large MEPs in the target limb are rewarded. Another BCI could aim to promote re-ownership of the limb affected by BID, as research has shown that ownership-like effects work similarly in people affected by BID and healthy controls (Lenggenhager et al., 2015). Lastly, there is a large body of literature on how specific patterns of EEG activity can be modified to lessen symptoms (Banerjee & Argáez, 2017; van der Kolk et al., 2016; Mennella et al., 2017; Qian et al., 2018). BID likely has similar patterns of activity that could be targeted, as neural correlates can be found in people with BID (Saetta et al., 2020).

An additional complexity of BID is that some brain areas, like the insula, are situated deeper in the brain than is readily accessible for EEG or TMS. This means that BCI systems for BID may need modalities that can handle the depths of the brain, like Magnetic Resonance Imaging (MRI).

8.6 Motor Inhibition and Falling in Elderly People

In the seventh chapter of this thesis titled “Structural Brain Connectivity in Elderly Individuals Relating to Motor Inhibition Networks” and the corresponding publication “Stopping Networks and Falling in the Elderly: Relating White Matter Architecture in the Brain’s Inhibitory Control Networks to Real-World Metrics of Falling in Elderly Adults” we investigated the research gap between real world behavioural data (falls), and white matter structural integrity between the inhibition networks (medical imaging data).

We found that higher microstructural integrity as measured with Apparent Fibre Density (AFD) in white matter pathways connecting the right inferior frontal gyrus and right subthalamic nucleus was associated with falling in older adults. We further investigated this counter intuitive result and found that physical activity influenced this relationship. The higher odds of falling were only found in combination with low levels of physical activity. This unexpected result adds to a growing body of literature showing that an age related decline observed with values such as Fractional Anisotropy (FA) may not correspond with new measures of brain derived data such as AFD (Choy et al., 2020).

While this project does not fit seamlessly within the BCI related aims of this thesis, some lessons can be drawn for BCI research. Knowing the strengths and limitations of other methods can help to design studies that investigate the working mechanisms of BCI that are not easily accessible to EEG or other BCI-friendly technologies. Further, reading literature from other fields, such as MRI or BID, may lead to innovative ideas by applying concepts of one research field to another. Finally, the knowledge gained in this work can help in designing a BCI that reduces falls in the elderly people.

8.7 Context

This work has advanced the field of medical BCIs in several important ways. We found that BCI lack scalability to make them financially effective, and justifiability to argue for their use in the first place. We suggested that the use and design of more specialised BCI may improve justifiability by increasing the ability to target specific mechanisms or problems in rehabilitation. We argued that this would automatically lead to more homogenised BCI, increasing scalability as designs converge on mechanisms that work well – which in turn will increase justifiability.

We followed through on the concept of more specialised BCI and in two studies created and tested the feasibility of the blueprint for a three-phase, poststroke rehabilitative BCI. In its first phase the BCI uses TMS-NF to teach participants to increase their corticospinal excitability using MI instructions, and in the last phase the participant can train, unsupervised, at home at their own pace using a much simpler EEG BCI. These phases could match the progression of stroke recovery much more closely than traditional physical therapy, for which considerable residual movement is necessary.

In our studies with healthy participants, the multimodal and multiphase BCI proved more efficient than the monomodal monophasic one. Not only did participants learn to control the BCI faster and better, they also showed a different quality of MI that was more

similar to actual ME than the control group. Therefore, we not only showed that the multimodal and multiphasic approach was more efficient and scalable, but we further cemented the justifiability of this specific MI for post stroke rehabilitation.

Once the medical justifiability of this BCI has been shown in a clinical population there are several optimisation steps that could render it more efficient and scalable. Transfer learning or alternative calibration procedures could significantly reduce the time and effort needed from a stroke survivor during the calibration phase, increasing the time and energy spent on training (Arvaneh et al., 2017; Azab et al., 2018, 2019). Further, the BCI could be improved to recognise errors or inattention from the user during trials (Martel et al., 2016; Wirth et al., 2019) which should increase speed and accuracy of the BCI use. Lastly, by adopting a user centred design perspective (Jeunet et al., 2020) improvements for every aspect of the human experience of using a BCI, from the instructions to the presentation of feedback (Roc et al., 2021), could further increase the efficiency. The medical justifiability and scalability of this BCI design builds a very strong case for more specialised, multimodal and multiphasic BCI design that is not restricted to applications only in poststroke motor rehabilitation.

We then synthesised the design principles of the three-phase BCI and elaborated three key points (neural signal, therapeutic goal, appropriate technology) to help BCI design stay focused on increasing scalability, justifiability and the therapeutic goal. Further, we took this approach and investigated a rare disorder called BID and after summarising the disorder, suggested several neural signals that could be targeted by BCI-based therapies. Lastly, instead of looking at a specific disease or disorder, we looked at the method of MRI and how it could contribute to the design of multimodal, multiphase BCI.

8.8 Strength and Limitations

Our experimental design stands out through the high number of participants and time-intensive repeated measures designs in our studies that are rare in contemporary literature. Further, thanks to careful design and extensive piloting, we experienced little attrition of equipment or participants during the experiments. We also controlled for known influencing variables and balanced our groups.

Our data processing stands out through the standard measures and processing pipelines, and the use of unbiased algorithms (that we supervised) whenever possible. We have relied on powerful and robust statistical analysis methods, and checked model or statistical test assumptions (and reported them). We further dissected our results, making

sure our interpretations were robust. We also preregistered the statistical analysis of our main publication. Additionally, most findings here represent additions of knowledge that are built upon and supported by the current literature. Lastly, the majority of the work presented in this thesis is (or in the process of being) peer reviewed – increasing its robustness.

While most findings reported in this body of work is incremental, the result is much stronger than the sum of its parts. We have played a part in shifting the discussion around medical BCI from a basic research technology perspective, to a translational approach to designing therapies (Colucci et al., 2022; Losanno et al., 2023; Remsik et al., 2022) with possible applications for home and wearable BCIs (Arpaia et al., 2023; Saibene et al., 2023). The publication of chapter three, five and seven are likely to increase the impact of our previous work, as they present more supporting data.

However, there are some limitations to keep in mind. Multimodal and multiphase BCI are by definition complex systems. Therefore, disentangling effects of different systems or even different timings of transition from one to the other, may be difficult. Further, collecting data with multiple systems creates complex datasets that can be analysed from multiple points of view. The rich data we collected may therefore contain additional information that is not yet analysed. Lastly, the data were collected using healthy participants, but most of our implications focus on BCI design for rehabilitative purposes. It is therefore likely that a clinical population might experience translational effects – however, a hospital pilot for TMS-NF was promising (Liang et al., 2020).

8.9 Outlook

An important next step is to implement multimodal and multiphase BCI in a clinical setting. Especially since the three-phase design presented in chapter three and four showed that the MI with TMS-NF might activate similar regions as ME, offering strong medical validity for this type of BCI.

It would also make sense to adapt this BCI approach to other medical targets, such as other motor rehabilitation challenges or rare diseases and disorders like BID or Functional Neurological Disorders. When widening the field for this BCI approach it should also be remembered that while TMS-NF has shown much promise, there are many other methods of noninvasive brainstimulation (e.g. Potok, 2022), many BCI adjacent research methods like MRI, and many research fields with research gaps that could profit from a BCI perspective (e.g. Ramduny et al., 2022), and experimental designs (e.g. Weijs et al., 2022) that could be used to address medical challenges using BCIs.

Finally, when a BCI system will prove successful and efficient for a medical challenge, there should be a push to converge on its working mechanism, and make it accessible to health professionals or individuals.

8.10 Conclusion

We have elaborated why current BCI lack scalability and justifiability, and we suggested potential solutions. Following our own advice, we have shown that a possible bimodal, three-phase poststroke rehabilitative BCI is not only feasible, but also addresses *scalability and justifiability*. We then distilled the design aspects into three key points. Additionally, we generalised our findings and looked at other disease areas and other methods suggesting possible applications and BCI-based therapies. Therefore, we have shown a new approach to BCI design centred around solving individual medical challenges and have shown its potential to change the landscape of medical BCI.

9. References

- Abdullahi, A., Van Crielkinge, T., Umar, N. A., Zakari, U. U., Truijen, S., & Saeys, W. (2021). Effect of constraint-induced movement therapy on persons-reported outcomes of health status after stroke: A systematic review and meta-analysis. *International Journal of Rehabilitation Research*, 44(1), 15. <https://doi.org/10.1097/MRR.0000000000000446>
- Abenna, S., Nahid, M., Bouyghf, H., & Ouacha, B. (2023). An enhanced motor imagery EEG signals prediction system in real-time based on delta rhythm. *Biomedical Signal Processing and Control*, 79, 104210. <https://doi.org/10.1016/j.bspc.2022.104210>
- Abiri, R., Borhani, S., Sellers, E. W., Jiang, Y., & Zhao, X. (2019). A comprehensive review of EEG-based brain–computer interface paradigms. *Journal of Neural Engineering*, 16(1), 011001. <https://doi.org/10.1088/1741-2552/aaf12e>
- Ablin, P., Cardoso, J.-F., & Gramfort, A. (2017). Faster ICA under orthogonal constraint. *ArXiv:1711.10873 [Stat]*. <http://arxiv.org/abs/1711.10873>
- Adadi, A., & Berrada, M. (2018). Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6, 52138–52160. <https://doi.org/10.1109/ACCESS.2018.2870052>
- Aggarwal, S., & Chugh, N. (2022). Review of Machine Learning Techniques for EEG Based Brain Computer Interface. *Archives of Computational Methods in Engineering*, 29(5), 3001–3020. <https://doi.org/10.1007/s11831-021-09684-6>
- Ahn, M., Ahn, S., Hong, J. H., Cho, H., Kim, K., Kim, B. S., Chang, J. W., & Jun, S. C. (2013a). Gamma band activity associated with BCI performance: Simultaneous MEG/EEG study. *Frontiers in Human Neuroscience*, 7. <https://www.frontiersin.org/articles/10.3389/fnhum.2013.00848>
- Ahn, M., Cho, H., Ahn, S., & Jun, S. C. (2013b). High Theta and Low Alpha Powers May Be Indicative of BCI-Illiteracy in Motor Imagery. *PLOS ONE*, 8(11), e80886. <https://doi.org/10.1371/journal.pone.0080886>
- Allison, B. Z., & Neuper, C. (2010). Could Anyone Use a BCI? In D. S. Tan & A. Nijholt (Eds.), *Brain-Computer Interfaces: Applying our Minds to Human-Computer Interaction* (pp. 35–54). Springer. https://doi.org/10.1007/978-1-84996-272-8_3
- Amboni, M., Barone, P., & Hausdorff, J. M. (2013). Cognitive contributions to gait and falls: Evidence and implications. *Movement Disorders*, 28(11), 1520–1533. <https://doi.org/10.1002/mds.25674>

- Ambrose, A. F., Paul, G., & Hausdorff, J. M. (2013). Risk factors for falls among older adults: A review of the literature. *Maturitas*, 75(1), 51–61. <https://doi.org/10.1016/j.maturitas.2013.02.009>
- Amunts, K., & Zilles, K. (2012). Architecture and organizational principles of Broca's region. *Trends in Cognitive Sciences*, 16(8), 418–426. <https://doi.org/10.1016/j.tics.2012.06.005>
- Aron, A. R., Durston, S., Eagle, D. M., Logan, G. D., Stinear, C. M., & Stuphorn, V. (2007). Converging Evidence for a Fronto-Basal-Ganglia Network for Inhibitory Control of Action and Cognition. *The Journal of Neuroscience*, 27(44), 11860. <https://doi.org/10.1523/JNEUROSCI.3644-07.2007>
- Aron, A. R., & Poldrack, R. A. (2006). Cortical and Subcortical Contributions to Stop Signal Response Inhibition: Role of the Subthalamic Nucleus. *The Journal of Neuroscience*, 26(9), 2424. <https://doi.org/10.1523/JNEUROSCI.4682-05.2006>
- Aron, A. R., Robbins, T. W., & Poldrack, R. A. (2014). Inhibition and the right inferior frontal cortex: One decade on. *Trends in Cognitive Sciences*, 18(4), 177–185. <https://doi.org/10.1016/j.tics.2013.12.003>
- Arpaia, P., Coyle, D., Esposito, A., Natalizio, A., Parvis, M., Pesola, M., & Vallefucio, E. (2023). Paving the Way for Motor Imagery-Based Tele-Rehabilitation through a Fully Wearable BCI System. *Sensors*, 23(13), Article 13. <https://doi.org/10.3390/s23135836>
- Arvaneh, M., Guan, C., Ang, K. K., Ward, T. E., Chua, K. S. G., Kuah, C. W. K., Ephraim Joseph, G. J., Phua, K. S., & Wang, C. (2017). Facilitating motor imagery-based brain–computer interface for stroke patients using passive movement. *Neural Computing and Applications*, 28(11), 3259–3272. <https://doi.org/10.1007/s00521-016-2234-7>
- Arzy, S., Overney, L. S., Landis, T., & Blanke, O. (2006). Neural Mechanisms of Embodiment: Asomatognosia Due to Premotor Cortex Damage. *Archives of Neurology*, 63(7), 1022–1025. <https://doi.org/10.1001/archneur.63.7.1022>
- Atmaca, M., Bingol, I., Aydin, A., Yildirim, H., Okur, I., Yildirim, M. A., Mermi, O., & Gurok, M. G. (2010). Brain morphology of patients with body dysmorphic disorder. *Journal of Affective Disorders*, 123(1), 258–263. <https://doi.org/10.1016/j.jad.2009.08.012>
- Azab, A. M., Toth, J., Mihaylova, L. S., & Arvaneh, M. (2018). A review on transfer learning approaches in brain–computer interface. In *Signal Processing and Machine Learning for Brain-Machine Interfaces* (pp. 81–101). IET Digital Library. https://doi.org/10.1049/PBCE114E_ch5

- Azab, A., Mihaylova, L., Ang, K. K., & Arvaneh, M. (2019). Weighted transfer learning for improving motor imagery-based brain-computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(7), Article 7.
- Bai, Z., Fong, K. N. K., Zhang, J. J., Chan, J., & Ting, K. H. (2020). Immediate and long-term effects of BCI-based rehabilitation of the upper extremity after stroke: A systematic review and meta-analysis. *Journal of NeuroEngineering and Rehabilitation*, 17(1), 57. <https://doi.org/10.1186/s12984-020-00686-2>
- Balasubramanian, S., Garcia-Cossio, E., Birbaumer, N., Burdet, E., & Ramos-Murguialday, A. (2018). Is EMG a Viable Alternative to BCI for Detecting Movement Intention in Severe Stroke? *IEEE Transactions on Biomedical Engineering*, 65(12), 2790–2797. <https://doi.org/10.1109/TBME.2018.2817688>
- Banerjee, S., & Argáez, C. (2017). *Neurofeedback and Biofeedback for Mood and Anxiety Disorders: A Review of Clinical Effectiveness and Guidelines*. Canadian Agency for Drugs and Technologies in Health. <http://www.ncbi.nlm.nih.gov/books/NBK531603/>
- Baniqued, P. D. E., Stanyer, E. C., Awais, M., Alazmani, A., Jackson, A. E., Mon-Williams, M. A., Mushtaq, F., & Holt, R. J. (2021). Brain–computer interface robotics for hand rehabilitation after stroke: A systematic review. *Journal of NeuroEngineering and Rehabilitation*, 18(1), Article 1. <https://doi.org/10.1186/s12984-021-00820-8>
- Barios, J. A., Ezquerro, S., Bertomeu-Motos, A., Nann, M., Badesa, Fco. J., Fernandez, E., Soekadar, S. R., & Garcia-Aracil, N. (2019). Synchronization of Slow Cortical Rhythms During Motor Imagery-Based Brain–Machine Interface Control. *International Journal of Neural Systems*, 29(05), 1850045. <https://doi.org/10.1142/S0129065718500454>
- Barrow, E., & Oyeboode, F. (2019). Body integrity identity disorder: Clinical features and ethical dimensions. *BJPsych Advances*, 25(3), 187–195. <https://doi.org/10.1192/bja.2018.55>
- Bassetti, C. L. A., Endres, M., Sander, A., Crean, M., Subramaniam, S., Carvalho, V., Di Liberto, G., Franco, O. H., Pijnenburg, Y., Leonardi, M., & Boon, P. (2022). The European Academy of Neurology Brain Health Strategy: One brain, one life, one approach. *European Journal of Neurology*, 29(9), 2559–2566. <https://doi.org/10.1111/ene.15391>
- Bavelier, D., Bediou, B., & Green, C. S. (2018). Expertise and generalization: Lessons from action video games. *Current Opinion in Behavioral Sciences*, 20, 169–173. <https://doi.org/10.1016/j.cobeha.2018.01.012>

- Bayne, T., & Levy, N. (2005). Amputees By Choice: Body Integrity Identity Disorder and the Ethics of Amputation. *Journal of Applied Philosophy*, 22(1), 75–86. <https://doi.org/10.1111/j.1468-5930.2005.00293.x>
- Beuter, A., Balossier, A., Trofimchuk, S., & Volpert, V. (2018). Modeling of post-stroke stimulation of cortical tissue. *Mathematical Biosciences*, 305, 146–159. <https://doi.org/10.1016/j.mbs.2018.08.014>
- Beuter, A., Balossier, A., Vassal, F., Hemm, S., & Volpert, V. (2020). Cortical stimulation in aphasia following ischemic stroke: Toward model-guided electrical neuromodulation. *Biological Cybernetics*, 114(1), 5–21. <https://doi.org/10.1007/s00422-020-00818-w>
- Biasiucci, A., Leeb, R., Iturrate, I., Perdakis, S., Al-Khodairy, A., Corbet, T., Schnider, A., Schmidlin, T., Zhang, H., Bassolino, M., Viceic, D., Vuadens, P., Guggisberg, A. G., & Millán, J. d R. (2018). Brain-actuated functional electrical stimulation elicits lasting arm motor recovery after stroke. *Nature Communications*, 9(1), Article 1. <https://doi.org/10.1038/s41467-018-04673-z>
- Bigoni, C., Cadic-Melchior, A., Morishita, T., & Hummel, F. C. (2023). Optimization of phase prediction for brain-state dependent stimulation: A grid-search approach. *Journal of Neural Engineering*, 20(1), 016039. <https://doi.org/10.1088/1741-2552/acb1d8>
- Blanke, O., & Metzinger, T. (2009). Full-body illusions and minimal phenomenal selfhood. *Trends in Cognitive Sciences*, 13(1), 7–13. <https://doi.org/10.1016/j.tics.2008.10.003>
- Bloch, F., Thibaud, M., Dugué, B., Brèque, C., Rigaud, A.-S., & Kemoun, G. (2011). Psychotropic drugs and falls in the elderly people: Updated literature review and meta-analysis. *Journal of Aging and Health*, 23(2), 329–346. <https://doi.org/10.1177/0898264310381277>
- Blokland, Y., Vlek1, R., Karaman, B., Özin, F., Thijssen, D., Eijsvogels, T., Colier, W., Floor-Westerdijk, M., Bruhn, J., & Farquhar, J. (2012). Detection of event-related desynchronization during attempted and imagined movements in tetraplegics for brain switch control. *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 3967–3969. <https://doi.org/10.1109/EMBC.2012.6346835>
- Blom, R. M., Hennekam, R. C., & Denys, D. (2012). Body Integrity Identity Disorder. *PLoS ONE*, 7(4), e34702. <https://doi.org/10.1371/journal.pone.0034702>

- Blom, R. M., van der Wal, S. J., Vulink, N. C., & Denys, D. (2017). Role of Sexuality in Body Integrity Identity Disorder (BIID): A Cross-Sectional Internet-Based Survey Study. *The Journal of Sexual Medicine*, 14(8), 1028–1035. <https://doi.org/10.1016/j.jsxm.2017.06.004>
- Blom, R. M., Wingen, G. A. van, Wal, S. J. van der, Luigjes, J., Dijk, M. T. van, Scholte, H. S., & Denys, D. (2016). The Desire for Amputation or Paralyzation: Evidence for Structural Brain Anomalies in Body Integrity Identity Disorder (BIID). *PLOS ONE*, 11(11), e0165789. <https://doi.org/10.1371/journal.pone.0165789>
- Boisgontier, M. P., & Cheval, B. (2016). The anova to mixed model transition. *Neuroscience & Biobehavioral Reviews*, 68, 1004–1005. <https://doi.org/10.1016/j.neubiorev.2016.05.034>
- Bonassi, G., Biggio, M., Bisio, A., Ruggeri, P., Bove, M., & Avanzino, L. (2017). Provision of somatosensory inputs during motor imagery enhances learning-induced plasticity in human motor cortex. *Scientific Reports*, 7(1), Article 1. <https://doi.org/10.1038/s41598-017-09597-0>
- Bonnier, P. (2009). Asomatognosia P. Bonnier. L'aschématie. *Revue Neurol* 1905;13:605-9. *Epilepsy & Behavior*, 16(3), 401–403. <https://doi.org/10.1016/j.yebeh.2009.09.020>
- Bottini, G., Sedda, A., Ferrè, E. R., Invernizzi, P., Gandola, M., & Paulesu, E. (2009). Productive symptoms in right brain damage. *Current Opinion in Neurology*, 22(6), 589–593. <https://doi.org/10.1097/wco.0b013e328332c71d>
- Bouckaert, F., Dols, A., Emsell, L., De Winter, F.-L., Vansteelandt, K., Claes, L., Sunaert, S., Stek, M., Sienaert, P., & Vandenbulcke, M. (2016). Relationship Between Hippocampal Volume, Serum BDNF, and Depression Severity Following Electroconvulsive Therapy in Late-Life Depression. *Neuropsychopharmacology*, 41(11), Article 11. <https://doi.org/10.1038/npp.2016.86>
- Braam, A. W., Visser, S., Cath, D. C., & Hoogendijk, W. J. G. (2006). Investigation of the Syndrome of Apotemnophilia and Course of a Cognitive-Behavioural Therapy. *Psychopathology*, 39(1), 32–37. <https://doi.org/10.1159/000089661>
- Brain, W. R. (1941). VISUAL DISORIENTATION WITH SPECIAL REFERENCE TO LESIONS OF THE RIGHT CEREBRAL HEMISPHERE. *Brain*, 64(4), 244–272. <https://doi.org/10.1093/brain/64.4.244>
- Buch, E., Weber, C., Cohen, L. G., Braun, C., Dimyan, M. A., Ard, T., Mellinger, J., Caria, A., Soekadar, S., Fourkas, A., & Birbaumer, N. (2008). Think to move: A neuromagnetic brain-computer interface

- (BCI) system for chronic stroke. *Stroke*, 39(3), 910–917.
<https://doi.org/10.1161/STROKEAHA.107.505313>
- Burns, E. J., Arnold, T., & Bukach, C. M. (2019). P-curving the fusiform face area: Meta-analyses support the expertise hypothesis. *Neuroscience & Biobehavioral Reviews*, 104, 209–221.
<https://doi.org/10.1016/j.neubiorev.2019.07.003>
- Butt, D. A., Mamdani, M., Austin, P. C., Tu, K., Gomes, T., & Glazier, R. H. (2012). The risk of hip fracture after initiating antihypertensive drugs in the elderly. *Archives of Internal Medicine*, 172(22), 1739–1744. <https://doi.org/10.1001/2013.jamainternmed.469>
- Buxbaum, L. J., Varghese, R., Stoll, H., & Winstein, C. J. (2020). Predictors of Arm Nonuse in Chronic Stroke: A Preliminary Investigation. *Neurorehabilitation and Neural Repair*, 34(6), 512–522.
<https://doi.org/10.1177/1545968320913554>
- Campbell, G. B., & Matthews, J. T. (2010). An integrative review of factors associated with falls during post-stroke rehabilitation. *Journal of Nursing Scholarship*, 42(4), 395–404. <https://doi.org/10.1111/j.1547-5069.2010.01369.x>
- Carson, R. G., Nelson, B. D., Buick, A. R., Carroll, T. J., Kennedy, N. C., & Cann, R. M. (2013). Characterizing changes in the excitability of corticospinal projections to proximal muscles of the upper limb. *Brain Stimulation*, 6(5), 760–768. <https://doi.org/10.1016/j.brs.2013.01.016>
- Carvalho, R., Dias, N., & Cerqueira, J. J. (2019). Brain-machine interface of upper limb recovery in stroke patients rehabilitation: A systematic review. *Physiotherapy Research International*, 24(2). Scopus. <https://doi.org/10.1002/pri.1764>
- Cattaneo, G., Bartrés-Faz, D., Morris, T. P., Sánchez, J. S., Macià, D., Tarrero, C., Tormos, J. M., & Pascual-Leone, A. (2018). The Barcelona Brain Health Initiative: A Cohort Study to Define and Promote Determinants of Brain Health. *Frontiers in Aging Neuroscience*, 10. <https://www.frontiersin.org/articles/10.3389/fnagi.2018.00321>
- Cervera, M. A., Soekadar, S. R., Ushiba, J., Millán, J. D. R., Liu, M., Birbaumer, N., & Garipelli, G. (2018). Brain-computer interfaces for post-stroke motor rehabilitation: A meta-analysis. *Annals of Clinical and Translational Neurology*, 5(5), 651–663. Scopus. <https://doi.org/10.1002/acn3.544>

- Chang, C.-M., Lin, H.-F., & Chiang, H.-H. (2015). A study on the relationship between age and inpatient falls in Taiwan. *International Journal of Nursing Practice*, 21(5), 605–611. <https://doi.org/10.1111/ijn.12342>
- Chang, C.-Y., Hsu, S.-H., Pion-Tonachini, L., & Jung, T.-P. (2018). Evaluation of Artifact Subspace Reconstruction for Automatic EEG Artifact Removal. *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 1242–1245. <https://doi.org/10.1109/EMBC.2018.8512547>
- Chavarriaga, R., Fried-Oken, M., Kleih, S., Lotte, F., & Scherer, R. (2017). Heading for new shores! Overcoming pitfalls in BCI design. *Brain Computer Interfaces (Abingdon, England)*, 4(1–2), 60–73. <https://doi.org/10.1080/2326263X.2016.1263916>
- Chen, Y., Demnitz, N., Yamamoto, S., Yaffé, K., Lawlor, B., & Leroi, I. (2022). Defining brain health: A concept analysis. *International Journal of Geriatric Psychiatry*, 37(1). <https://doi.org/10.1002/gps.5564>
- Choy, S. W., Bagarinao, E., Watanabe, H., Ho, E. T. W., Maesawa, S., Mori, D., Hara, K., Kawabata, K., Yoneyama, N., Ohdake, R., Imai, K., Masuda, M., Yokoi, T., Ogura, A., Taoka, T., Koyama, S., Tanabe, H. C., Katsuno, M., Wakabayashi, T., ... Sobue, G. (2020). Changes in white matter fiber density and morphology across the adult lifespan: A cross-sectional fixel-based analysis. *Human Brain Mapping*, 41(12), 3198–3211. <https://doi.org/10.1002/hbm.25008>
- Cohen, M. X. (2014). *Analyzing Neural Time Series Data: Theory and Practice*. <https://doi.org/10.7551/mitpress/9609.001.0001>
- Cohen, R. G., Nutt, J. G., & Horak, F. B. (2011). Errors in Postural Preparation Lead to Increased Choice Reaction Times for Step Initiation in Older Adults. *The Journals of Gerontology: Series A*, 66A(6), 705–713. <https://doi.org/10.1093/gerona/qlr054>
- Colucci, A., Vermehren, M., Cavallo, A., Angerhöfer, C., Peekhaus, N., Zollo, L., Kim, W.-S., Paik, N.-J., & Soekadar, S. R. (2022). Brain–Computer Interface-Controlled Exoskeletons in Clinical Neurorehabilitation: Ready or Not? *Neurorehabilitation and Neural Repair*, 36(12), 747–756. <https://doi.org/10.1177/15459683221138751>
- Confalonieri, L., Pagnoni, G., Barsalou, L. W., Rajendra, J., Eickhoff, S. B., & Butler, A. J. (2012). Brain Activation in Primary Motor and Somatosensory Cortices during Motor Imagery Correlates with

- Motor Imagery Ability in Stroke Patients. *ISRN Neurology*, 2012, 1–17.
<https://doi.org/10.5402/2012/613595>
- Corbet, T., Iturrate, I., Pereira, M., Perdakis, S., & Millán, J. del R. (2018). Sensory threshold neuromuscular electrical stimulation fosters motor imagery performance. *NeuroImage*, 176, 268–276.
<https://doi.org/10.1016/j.neuroimage.2018.04.005>
- Corbetta, D., Sirtori, V., Castellini, G., Moja, L., & Gatti, R. (2015). Constraint-induced movement therapy for upper extremities in people with stroke. *Cochrane Database of Systematic Reviews*, 10.
<https://doi.org/10.1002/14651858.CD004433.pub3>
- Coscia, M., Wessel, M. J., Chaudary, U., Millán, J. D. R., Micera, S., Guggisberg, A., Vuadens, P., Donoghue, J., Birbaumer, N., & Hummel, F. C. (2019). Neurotechnology-aided interventions for upper limb motor rehabilitation in severe chronic stroke. *Brain*, 142(8), 2182–2197. Scopus.
<https://doi.org/10.1093/brain/awz181>
- Coxon, J. P., Van Impe, A., Wenderoth, N., & Swinnen, S. P. (2012). Aging and Inhibitory Control of Action: Cortico-Subthalamic Connection Strength Predicts Stopping Performance. *The Journal of Neuroscience*, 32(24), 8401. <https://doi.org/10.1523/JNEUROSCI.6360-11.2012>
- Craig, C. L., Marshall, A. L., Sjöström, M., Bauman, A. E., Booth, M. L., Ainsworth, B. E., Pratt, M., Ekelund, U., Yngve, A., Sallis, J. F., & Oja, P. (2003). International Physical Activity Questionnaire: 12-Country Reliability and Validity: *Medicine & Science in Sports & Exercise*, 35(8), 1381–1395.
<https://doi.org/10.1249/01.MSS.0000078924.61453.FB>
- Cremen, I. A., & Carson, R. G. (2017). Have Standard Tests of Cognitive Function Been Misappropriated in the Study of Cognitive Enhancement? *Frontiers in Human Neuroscience*, 11.
<https://doi.org/10.3389/fnhum.2017.00276>
- Critchley, M. (1954). The Parietal Lobes. *Neurology*, 4(5), 402–402. <https://doi.org/10.1212/WNL.4.5.402>
- Cruz, A., Pires, G., Lopes, A. C., & Nunes, U. J. (2019). Detection of Stressful Situations Using GSR While Driving a BCI-controlled Wheelchair. *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 1651–1656.
<https://doi.org/10.1109/EMBC.2019.8857748>
- de Azevedo, J. A., Barbosa, F. D. S., Seixas, V. M., da Silva Scipioni, K. R. D., Sampaio, P. Y. S., da Cruz, D. M. C., Piscitelli, D., Chui, K. K., & de Freitas Zanona, A. (2022). Effects of constraint-induced

- movement therapy on activity and participation after a stroke: Systematic review and meta-analysis. *Frontiers in Human Neuroscience*, 16. <https://www.frontiersin.org/articles/10.3389/fnhum.2022.987061>
- de Castro-Cros, M., Sebastian-Romagosa, M., Rodríguez-Serrano, J., Opisso, E., Ochoa, M., Ortner, R., Guger, C., & Tost, D. (2020). Effects of Gamification in BCI Functional Rehabilitation. *Frontiers in Neuroscience*, 14. <https://doi.org/10.3389/fnins.2020.00882>
- Deandrea, S., Lucenteforte, E., Bravi, F., Foschi, R., La Vecchia, C., & Negri, E. (2010). Risk Factors for Falls in Community-dwelling Older People: A Systematic Review and Meta-analysis. *Epidemiology*, 21(5), 658. <https://doi.org/10.1097/EDE.0b013e3181e89905>
- Decety, J. (1996). The neurophysiological basis of motor imagery. *Behavioural Brain Research*, 77(1), 45–52. [https://doi.org/10.1016/0166-4328\(95\)00225-1](https://doi.org/10.1016/0166-4328(95)00225-1)
- Dechent, P., Merboldt, K.-D., & Frahm, J. (2004). Is the human primary motor cortex involved in motor imagery? *Cognitive Brain Research*, 19(2), 138–144. <https://doi.org/10.1016/j.cogbrainres.2003.11.012>
- Dell’Acqua, F., & Tournier, J.-D. (2019). Modelling white matter with spherical deconvolution: How and why? *NMR in Biomedicine*, 32(4), e3945. <https://doi.org/10.1002/nbm.3945>
- Delorme, A., & Makeig, S. (2004). EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, 134(1), 9–21. <https://doi.org/10.1016/j.jneumeth.2003.10.009>
- Deng, W., Rolls, E. T., Ji, X., Robbins, T. W., Banaschewski, T., Bokde, A. L. W., Bromberg, U., Buechel, C., Desrivières, S., Conrod, P., Flor, H., Frouin, V., Gallinat, J., Garavan, H., Gowland, P., Heinz, A., Ittermann, B., Martinot, J.-L., Lemaitre, H., ... Feng, J. (2017). Separate neural systems for behavioral change and for emotional responses to failure during behavioral inhibition. *Human Brain Mapping*, 38(7), 3527–3537. <https://doi.org/10.1002/hbm.23607>
- Devanne, H., Lavoie, B. A., & Capaday, C. (1997). Input-output properties and gain changes in the human corticospinal pathway. *Experimental Brain Research*, 114(2), 329–338. <https://doi.org/10.1007/PL00005641>

- Di Rienzo, F., Collet, C., Hoyek, N., & Guillot, A. (2014). Impact of Neurologic Deficits on Motor Imagery: A Systematic Review of Clinical Evaluations. *Neuropsychology Review*, 24(2), 116–147. <https://doi.org/10.1007/s11065-014-9257-6>
- Diamond, A. (2013). Executive Functions. *Annual Review of Psychology*, 64(1), 135–168. <https://doi.org/10.1146/annurev-psych-113011-143750>
- Donoghue, O. A., McGarrigle, C. A., Foley, M., Fagan, A., Meaney, J., & Kenny, R. A. (2018). Cohort Profile Update: The Irish Longitudinal Study on Ageing (TILDA). *International Journal of Epidemiology*, 47(5), 1398–1398l. <https://doi.org/10.1093/ije/dyy163>
- Doody, R. S., & Jankovic, J. (1992). The alien hand and related signs. *Journal of Neurology, Neurosurgery, and Psychiatry*, 55(9), 806–810. <https://doi.org/10.1136/jnnp.55.9.806>
- Du, J., Rolls, E. T., Cheng, W., Li, Y., Gong, W., Qiu, J., & Feng, J. (2020). Functional connectivity of the orbitofrontal cortex, anterior cingulate cortex, and inferior frontal gyrus in humans. *Cortex*, 123, 185–199. <https://doi.org/10.1016/j.cortex.2019.10.012>
- Egorova, N., Dhollander Thijs, Khelif Mohamed Salah, Khan Wasim, Werden Emilio, & Brodtmann Amy. (2020). Pervasive White Matter Fiber Degeneration in Ischemic Stroke. *Stroke*, 51(5), 1507–1513. <https://doi.org/10.1161/STROKEAHA.119.028143>
- England, D., Ruddy, K. L., Dakin, C. J., Schwartz, S. E., Butler, B., & Bolton, D. A. E. (2021). Relationship between Speed of Response Inhibition and Ability to Suppress a Step in Midlife and Older Adults. *Brain Sciences*, 11(5), Article 5. <https://doi.org/10.3390/brainsci11050643>
- Enz, N., Ruddy, K. L., Rueda-Delgado, L. M., & Whelan, R. (2021). Volume of β -Bursts, But Not Their Rate, Predicts Successful Response Inhibition. *The Journal of Neuroscience*, 41(23), 5069. <https://doi.org/10.1523/JNEUROSCI.2231-20.2021>
- Faul, F., Erdfelder, E., Lang, A.-G., & Buchner, A. (2007). G*Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior Research Methods*, 39(2), 175–191. <https://doi.org/10.3758/bf03193146>
- Fazli, S., Mehnert, J., Steinbrink, J., Curio, G., Villringer, A., Müller, K.-R., & Blankertz, B. (2012). Enhanced performance by a hybrid NIRS–EEG brain computer interface. *NeuroImage*, 59(1), 519–529. <https://doi.org/10.1016/j.neuroimage.2011.07.084>

- Feigin, V. L., Stark, B. A., Johnson, C. O., Roth, G. A., Bisignano, C., Abady, G. G., Abbasifard, M., Abbasi-Kangevari, M., Abd-Allah, F., Abedi, V., Abualhasan, A., Abu-Rmeileh, N. M., Abushouk, A. I., Adebayo, O. M., Agarwal, G., Agasthi, P., Ahinkorah, B. O., Ahmad, S., Ahmadi, S., ... Murray, C. J. L. (2021). Global, regional, and national burden of stroke and its risk factors, 1990–2019: A systematic analysis for the Global Burden of Disease Study 2019. *The Lancet Neurology*, 20(10), 795–820. [https://doi.org/10.1016/S1474-4422\(21\)00252-0](https://doi.org/10.1016/S1474-4422(21)00252-0)
- Feinberg, T. E., Schindler, R. J., Flanagan, N. G., & Haber, L. D. (1992). Two alien hand syndromes. *Neurology*, 42(1), 19–24. <https://doi.org/10.1212/wnl.42.1.19>
- Fernandez Guerrero, A., & Achermann, P. (2019). Brain dynamics during the sleep onset transition: An EEG source localization study. *Neurobiology of Sleep and Circadian Rhythms*, 6, 24–34. <https://doi.org/10.1016/j.nbscr.2018.11.001>
- Fiebelkorn, I. C., & Kastner, S. (2019). A Rhythmic Theory of Attention. *Trends in Cognitive Sciences*, 23(2), 87–101. <https://doi.org/10.1016/j.tics.2018.11.009>
- First, M. (2005). Desire for amputation of a limb: Paraphilia, psychosis, or a new type of identity disorder. *Psychological Medicine*, 35, 919–928. <https://doi.org/10.1017/S0033291704003320>
- First, M. B., & Fisher, C. E. (2012). Body Integrity Identity Disorder: The Persistent Desire to Acquire a Physical Disability. *Psychopathology*, 45(1), 3–14. <https://doi.org/10.1159/000330503>
- Foldes, S. T., Weber, D. J., & Collinger, J. L. (2015). MEG-based neurofeedback for hand rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 12(1), 85. <https://doi.org/10.1186/s12984-015-0076-7>
- Fornaro, S., Patrikelis, P., & Lucci, G. (2020). When having a limb means feeling overcomplete. Xenomelia, the chronic sense of disownership and the right parietal lobe hypothesis. *Laterality*, 0(0), 1–20. <https://doi.org/10.1080/1357650X.2020.1866000>
- Franse, C. B., Rietjens, J. A., Burdorf, A., van Grieken, A., Korfage, I. J., van der Heide, A., Raso, F. M., van Beeck, E., & Raat, H. (2017). A prospective study on the variation in falling and fall risk among community-dwelling older citizens in 12 European countries. *BMJ Open*, 7(6), e015827. <https://doi.org/10.1136/bmjopen-2017-015827>
- Fuster, J. M. (2008). *The Prefrontal Cortex (Fourth Edition)* (J. M. Fuster, Ed.). Academic Press. <https://doi.org/10.1016/B978-0-12-373644-4.00001-3>

- Gabard-Durnam, L. J., Mendez Leal, A. S., Wilkinson, C. L., & Levin, A. R. (2018). The Harvard Automated Processing Pipeline for Electroencephalography (HAPPE): Standardized Processing Software for Developmental and High-Artifact Data. *Frontiers in Neuroscience*, 12. <https://doi.org/10.3389/fnins.2018.00097>
- Gandola, M., Invernizzi, P., Sedda, A., Ferrè, E. R., Sterzi, R., Sberna, M., Paulesu, E., & Bottini, G. (2012). An anatomical account of somatoparaphrenia. *Cortex; a Journal Devoted to the Study of the Nervous System and Behavior*, 48(9), 1165–1178. <https://doi.org/10.1016/j.cortex.2011.06.012>
- Gangwani, R., Cain, A., Collins, A., & Cassidy, J. M. (2022). Leveraging Factors of Self-Efficacy and Motivation to Optimize Stroke Recovery. *Frontiers in Neurology*, 13. <https://www.frontiersin.org/articles/10.3389/fneur.2022.823202>
- Garcia-Falgueras, A., Madrid, & Spain. (2014). Gender Dysphoria and Body Integrity Identity Disorder: Similarities and Differences. *Psychology*, 05(02), Article 02. <https://doi.org/10.4236/psych.2014.52025>
- Gerstmann, J. (1942). PROBLEM OF IMPERCEPTION OF DISEASE AND OF IMPAIRED BODY TERRITORIES WITH ORGANIC LESIONS: RELATION TO BODY SCHEME AND ITS DISORDERS. *Archives of Neurology & Psychiatry*, 48(6), 890–913. <https://doi.org/10.1001/archneurpsyc.1942.02290120042003>
- Ghaziri, J., Tucholka, A., Larue, V., Blanchette-Sylvestre, M., Reyburn, G., Gilbert, G., Lévesque, J., & Beauregard, M. (2013). Neurofeedback Training Induces Changes in White and Gray Matter. *Clinical EEG and Neuroscience*, 44(4), 265–272. <https://doi.org/10.1177/1550059413476031>
- Giummarra, M. J., Bradshaw, J. L., Nicholls, M. E. R., Hilti, L. M., & Brugger, P. (2011). Body Integrity Identity Disorder: Deranged Body Processing, Right Fronto-Parietal Dysfunction, and Phenomenological Experience of Body Incongruity. *Neuropsychology Review*, 21(4), 320–333. <https://doi.org/10.1007/s11065-011-9184-8>
- Golaszewski, S., Frey, V., Thomschewski, A., Sebastianelli, L., Versace, V., Saltuari, L., Trinka, E., & Nardone, R. (2021). Neural mechanisms underlying the Rubber Hand Illusion: A systematic review of related neurophysiological studies. *Brain and Behavior*, 11(8), e02124. <https://doi.org/10.1002/brb3.2124>
- Grabherr, L., Macaudo, G., & Lenggenhager, B. (2015). The Moving History of Vestibular Stimulation as a Therapeutic Intervention. *Multisensory Research*, 28(5–6), 653–687. <https://doi.org/10.1163/22134808-00002495>

- Green, C. S., & Bavelier, D. (2008). Exercising your brain: A review of human brain plasticity and training-induced learning. *Psychology and Aging, 23*(4), 692–701. <https://doi.org/10.1037/a0014345>
- Grivaz, P., Blanke, O., & Serino, A. (2017). Common and distinct brain regions processing multisensory bodily signals for peripersonal space and body ownership. *NeuroImage, 147*, 602–618. <https://doi.org/10.1016/j.neuroimage.2016.12.052>
- Grochowski, A., Bücken, C., Kasten, E., Osen, B., & Heinrichs, N. (2018). Mental Images in Body Integrity Identity Disorder (BIID) and in Body Dysmorphic Disorder (BDD): An Exploratory Study. *Behaviour Change, 35*(3), 174–184. <https://doi.org/10.1017/bec.2018.17>
- Gueorguieva, R., & Krystal, J. H. (2004). Move Over ANOVA: Progress in Analyzing Repeated-Measures Data and Its Reflection in Papers Published in the Archives of General Psychiatry. *Archives of General Psychiatry, 61*(3), 310–317. <https://doi.org/10.1001/archpsyc.61.3.310>
- Ha, V.-A. T., Nguyen, T. N., Nguyen, T. X., Nguyen, H. T. T., Nguyen, T. T. H., Nguyen, A. T., Pham, T., & Vu, H. T. T. (2021). Prevalence and Factors Associated with Falls among Older Outpatients. *International Journal of Environmental Research and Public Health, 18*(8), Article 8. <https://doi.org/10.3390/ijerph18084041>
- Hänggi, J., Bellwald, D., & Brugger, P. (2016). Shape alterations of basal ganglia and thalamus in xenomelia. *NeuroImage: Clinical, 11*, 760–769. <https://doi.org/10.1016/j.nicl.2016.05.015>
- Härlein, J., Dassen, T., Halfens, R. J. G., & Heinze, C. (2009). Fall risk factors in older people with dementia or cognitive impairment: A systematic review. *Journal of Advanced Nursing, 65*(5), 922–933. <https://doi.org/10.1111/j.1365-2648.2008.04950.x>
- Harmony, T. (2013). The functional significance of delta oscillations in cognitive processing. *Frontiers in Integrative Neuroscience, 7*. <https://www.frontiersin.org/articles/10.3389/fnint.2013.00083>
- Hartikainen, S., Lönnroos, E., & Louhivuori, K. (2007). Medication as a risk factor for falls: Critical systematic review. *The Journals of Gerontology. Series A, Biological Sciences and Medical Sciences, 62*(10), 1172–1181. <https://doi.org/10.1093/gerona/62.10.1172>
- Hasegawa, Y., Kasai, T., Kinoshita, H., & Yahagi, S. (2001). Modulation of a motor evoked response to transcranial magnetic stimulation by the activity level of the first dorsal interosseous muscle in humans when grasping a stationary object with different grip widths. *Neuroscience Letters, 299*(1), 1–4. [https://doi.org/10.1016/S0304-3940\(00\)01709-2](https://doi.org/10.1016/S0304-3940(00)01709-2)

- Hatem, S. M., Saussez, G., della Faille, M., Zhang, X., Dispa, D., & Bleyenheuft, Y. (2016). Rehabilitation of motor function after stroke: A multiple systematic review focused on techniques to stimulate upper extremity recovery. *Frontiers in Human Neuroscience* 2016 Sep 13;10(442):Epub. Downloaded from PEDro <http://www.pedro.org.au>.
- Hawe, R. L., Scott, S. H., & Dukelow, S. P. (2019). Taking Proportional Out of Stroke Recovery. *Stroke*, 50(1), 204–211. <https://doi.org/10.1161/STROKEAHA.118.023006>
- Herman, T., Mirelman, A., Giladi, N., Schweiger, A., & Hausdorff, J. M. (2010). Executive Control Deficits as a Prodrome to Falls in Healthy Older Adults: A Prospective Study Linking Thinking, Walking, and Falling. *The Journals of Gerontology: Series A*, 65A(10), 1086–1092. <https://doi.org/10.1093/gerona/gdq077>
- Hess, C. W., Mills, K. R., & Murray, N. M. F. (1986). Magnetic stimulation of the human brain: Facilitation of motor responses by voluntary contraction of ipsilateral and contralateral muscles with additional observations on an amputee. *Neuroscience Letters*, 71(2), 235–240. [https://doi.org/10.1016/0304-3940\(86\)90565-3](https://doi.org/10.1016/0304-3940(86)90565-3)
- Hétu, S., Grégoire, M., Saimpont, A., Coll, M.-P., Eugène, F., Michon, P.-E., & Jackson, P. L. (2013). The neural network of motor imagery: An ALE meta-analysis. *Neuroscience & Biobehavioral Reviews*, 37(5), 930–949. <https://doi.org/10.1016/j.neubiorev.2013.03.017>
- Hilti, L. M., Hänggi, J., Vitacco, D. A., Kraemer, B., Palla, A., Luechinger, R., Jäncke, L., & Brugger, P. (2013). The desire for healthy limb amputation: Structural brain correlates and clinical features of xenomelia. *Brain*, 136(1), 318–329. <https://doi.org/10.1093/brain/aws316>
- Hong, X., Lu, Z. K., Teh, I., Nasrallah, F. A., Teo, W. P., Ang, K. K., Phua, K. S., Guan, C., Chew, E., & Chuang, K.-H. (2017). Brain plasticity following MI-BCI training combined with tDCS in a randomized trial in chronic subcortical stroke subjects: A preliminary study. *Scientific Reports*, 7(1), Article 1. <https://doi.org/10.1038/s41598-017-08928-5>
- Hope, T. M. H., Friston, K., Price, C. J., Leff, A. P., Rotshtein, P., & Bowman, H. (2019). Recovery after stroke: Not so proportional after all? *Brain*, 142(1), 15–22. <https://doi.org/10.1093/brain/awy302>
- Ishii, R., Canuet, L., Ishihara, T., Aoki, Y., Ikeda, S., Hata, M., Katsimichas, T., Gunji, A., Takahashi, H., Nakahachi, T., Iwase, M., & Takeda, M. (2014). Frontal midline theta rhythm and gamma power changes during focused attention on mental calculation: An MEG beamformer analysis. *Frontiers in Human Neuroscience*, 8. <https://www.frontiersin.org/articles/10.3389/fnhum.2014.00406>

- Jana, S., Hannah, R., Muralidharan, V., & Aron, A. R. (2020). Temporal cascade of frontal, motor and muscle processes underlying human action-stopping. *ELife*, 9, e50371. <https://doi.org/10.7554/eLife.50371>
- Jenkinson, P. M., Edelmystyn, N. M. J., Preston, C., & Ellis, S. J. (2015). Anarchic hand with abnormal agency following right inferior parietal lobe damage: A case report. *Neurocase*, 21(4), 471–478. <https://doi.org/10.1080/13554794.2014.925936>
- Jeunet, C., Benaroch, C., Cabestaing, F., Chavarriaga, R., Colamarino, E., Corsi, M.-C., Coyle, D., De Vico Fallani, F., Enriquez-Geppert, S., Figueirédo, P., Grosse-Wentrup, M., Kleih, S., Kober, S., Kübler, A., Lotte, F., Maby, E., Mattia, D., Mattout, J., Müller-Putz, G. R., ... Wriessnegger, S. (2020, September 29). *A user-centred approach to unlock the potential of non-invasive BCIs: An unprecedented international translational effort*. CHIST-ERA Conference 2020, Virtual Conference. <https://www.chistera.eu/conference/2020/abstract/camille-jeunet-0>
- Jeunet, C., Jahanpour, E., & Lotte, F. (2016). Why standard brain-computer interface (BCI) training protocols should be changed: An experimental study. *Journal of Neural Engineering*, 13(3), 036024. <https://doi.org/10.1088/1741-2560/13/3/036024>
- Käthner, I., Halder, S., Hintermüller, C., Espinosa, A., Guger, C., Miralles, F., Vargiu, E., Dauwalder, S., Rafael-Palou, X., Solà, M., Daly, J. M., Armstrong, E., Martin, S., & Kübler, A. (2017). A Multifunctional Brain-Computer Interface Intended for Home Use: An Evaluation with Healthy Participants and Potential End Users with Dry and Gel-Based Electrodes. *Frontiers in Neuroscience*, 11. <https://doi.org/10.3389/fnins.2017.00286>
- Kaiser, V., Kreiling, A., Mueller-Putz, G., & Neuper, C. (2011). First Steps Toward a Motor Imagery Based Stroke BCI: New Strategy to Set up a Classifier. *Frontiers in Neuroscience*, 5, 86. <https://doi.org/10.3389/fnins.2011.00086>
- Karabanov, A. N., Madsen, K. H., Krohne, L. G., & Siebner, H. R. (2021). Does pericentral mu-rhythm “power” corticomotor excitability? – A matter of EEG perspective. *Brain Stimulation*, 14(3), 713–722. <https://doi.org/10.1016/j.brs.2021.03.017>
- Karlsson, M. K., Magnusson, H., von Schewelov, T., & Rosengren, B. E. (2013). Prevention of falls in the elderly—A review. *Osteoporosis International*, 24(3), 747–762. <https://doi.org/10.1007/s00198-012-2256-7>
- Kaur, B., Singh, D., & Roy, P. P. (2019). Age and gender classification using brain–computer interface. *Neural Computing and Applications*, 31(10), 5887–5900. <https://doi.org/10.1007/s00521-018-3397-1>

- Kearney, F. C., Harwood, R. H., Gladman, J. R. F., Lincoln, N., & Masud, T. (2013). The relationship between executive function and falls and gait abnormalities in older adults: A systematic review. *Dementia and Geriatric Cognitive Disorders*, 36(1–2), 20–35. <https://doi.org/10.1159/000350031>
- Kerous, B., Skola, F., & Liarokapis, F. (2018). EEG-based BCI and video games: A progress report. *Virtual Reality*, 22(2), 119–135. <https://doi.org/10.1007/s10055-017-0328-x>
- Keute, M., Gebühr, J.-S., Guggenberger, R., Trunk, B. H., & Gharabaghi, A. (2023). *Phase-specific stimulation reveals consistent sinusoidal modulation of human corticospinal excitability along the oscillatory beta cycle* (p. 2023.04.25.538229). bioRxiv. <https://doi.org/10.1101/2023.04.25.538229>
- Khalatbari-Soltani, S., Stanaway, F., Sherrington, C., Blyth, F. M., Naganathan, V., Handelsman, D. J., Seibel, M. J., Waite, L. M., Le Couteur, D. G., & Cumming, R. G. (2021). The Prospective Association Between Socioeconomic Status and Falls Among Community-Dwelling Older Men. *The Journals of Gerontology. Series A, Biological Sciences and Medical Sciences*, 76(10), 1821–1828. <https://doi.org/10.1093/gerona/glab038>
- Khan, M. A., Das, R., Iversen, H. K., & Puthusserypady, S. (2020). Review on motor imagery based BCI systems for upper limb post-stroke neurorehabilitation: From designing to application. *Computers in Biology and Medicine*, 123, 103843. <https://doi.org/10.1016/j.compbiomed.2020.103843>
- Kotajima, F., Meadows, G. E., Morrell, M. J., & Corfield, D. R. (2005). Cerebral blood flow changes associated with fluctuations in alpha and theta rhythm during sleep onset in humans. *The Journal of Physiology*, 568(1), 305–313. <https://doi.org/10.1113/jphysiol.2005.092577>
- Kovyazina, M. S., Varako, N. A., Lyukmanov, R. K., Asiatskaya, G. A., Suponeva, N. A., & Trofimova, A. K. (2019). Neurofeedback in the Rehabilitation of Patients with Motor Disorders after Stroke. *Human Physiology*, 45(4), 444–451. Scopus. <https://doi.org/10.1134/S0362119719040042>
- Krakauer, J. W., & Carmichael, S. T. (2017). *Broken Movement: The Neurobiology of Motor Recovery After Stroke*. MIT Press.
- Kröger, K., Schnell, T., & Kasten, E. (2014). Effects of psychotherapy on patients suffering from Body Integrity Identity Disorder (BIID). *American Journal of Applied Psychology*, 6. <https://doi.org/10.11648/j.ajap.20140305.11>
- Kübler, A., Holz, E. M., Riccio, A., Zickler, C., Kaufmann, T., Kleih, S. C., Staiger-Sälzer, P., Desideri, L., Hoogerwerf, E.-J., & Mattia, D. (2014). The User-Centered Design as Novel Perspective for

- Evaluating the Usability of BCI-Controlled Applications. *PLoS ONE*, 9(12).
<https://doi.org/10.1371/journal.pone.0112392>
- Kwakkel, G., Veerbeek, J. M., van Wegen, E. E. H., & Wolf, S. L. (2015). Constraint-Induced Movement Therapy after Stroke. *The Lancet. Neurology*, 14(2), 224–234. [https://doi.org/10.1016/S1474-4422\(14\)70160-7](https://doi.org/10.1016/S1474-4422(14)70160-7)
- Lamb, S. E., Ferrucci, L., Volapto, S., Fried, L. P., Guralnik, J. M., & Women's Health and Aging Study. (2003). Risk factors for falling in home-dwelling older women with stroke: The Women's Health and Aging Study. *Stroke*, 34(2), 494–501. <https://doi.org/10.1161/01.STR.0000053444.00582.B7>
- Lang, C. E., Lohse, K. R., & Birkenmeier, R. L. (2015). Dose and timing in neurorehabilitation: Prescribing motor therapy after stroke. *Current Opinion in Neurology*, 28(6), 549–555.
<https://doi.org/10.1097/WCO.0000000000000256>
- Langhorne, P., Bernhardt, J., & Kwakkel, G. (2011). Stroke rehabilitation. *The Lancet*, 377(9778), 1693–1702.
[https://doi.org/10.1016/S0140-6736\(11\)60325-5](https://doi.org/10.1016/S0140-6736(11)60325-5)
- Lawrence, A. A. (2010). Sexual Orientation versus Age of Onset as Bases for Typologies (Subtypes) for Gender Identity Disorder in Adolescents and Adults. *Archives of Sexual Behavior*, 39(2), 514–545.
<https://doi.org/10.1007/s10508-009-9594-3>
- Lawrence, E. S., Coshall Catherine, Dundas Ruth, Stewart Judy, Rudd Anthony G., Howard Robin, & Wolfe Charles D. A. (2001). Estimates of the Prevalence of Acute Stroke Impairments and Disability in a Multiethnic Population. *Stroke*, 32(6), 1279–1284. <https://doi.org/10.1161/01.STR.32.6.1279>
- Leamy, D. J., Collins, R., & Ward, T. E. (2011). Combining fNIRS and EEG to Improve Motor Cortex Activity Classification during an Imagined Movement-Based Task. In D. D. Schmorow & C. M. Fidopiastis (Eds.), *Foundations of Augmented Cognition. Directing the Future of Adaptive Systems* (pp. 177–185). Springer. https://doi.org/10.1007/978-3-642-21852-1_22
- Lebon, F., Byblow, W. D., Collet, C., Guillot, A., & Stinear, C. M. (2012). The modulation of motor cortex excitability during motor imagery depends on imagery quality. *European Journal of Neuroscience*, 35(2), 323–331. <https://doi.org/10.1111/j.1460-9568.2011.07938.x>
- Lee, P. H., Macfarlane, D. J., Lam, T., & Stewart, S. M. (2011). Validity of the international physical activity questionnaire short form (IPAQ-SF): A systematic review. *International Journal of Behavioral Nutrition and Physical Activity*, 8(1), 115. <https://doi.org/10.1186/1479-5868-8-115>

- Leemans, A., Jeurissen, B., Sijbers, J., & Jones, D. K. (2009). *ExploreDTI: a graphical tool- box for processing, analyzing and visualizing diffusion MR data*. <https://archive.ismrm.org/2009/3537.html>
- Leemans, A., & Jones, D. K. (2009). The B-matrix must be rotated when correcting for subject motion in DTI data. *Magnetic Resonance in Medicine*, *61*(6), 1336–1349. <https://doi.org/10.1002/mrm.21890>
- Lenggenhager, B., Hilti, L., & Brugger, P. (2015). Disturbed body integrity and the “rubber foot illusion.” *Neuropsychology*, *29*(2), 205–211. <https://doi.org/10.1037/neu0000143>
- Lenggenhager, B., Hilti, L., Palla, A., Macaudo, G., & Brugger, P. (2014). Vestibular stimulation does not diminish the desire for amputation. *Cortex*, *54*, 210–212. <https://doi.org/10.1016/j.cortex.2014.02.004>
- Leonardi, G., Ciurleo, R., Cucinotta, F., Fonti, B., Borzelli, D., Costa, L., Tisano, A., Portaro, S., & Alito, A. (2022). The role of brain oscillations in post-stroke motor recovery: An overview. *Frontiers in Systems Neuroscience*, *16*, 947421. <https://doi.org/10.3389/fnsys.2022.947421>
- Li, K. Z. H., Bherer, L., Mirelman, A., Maidan, I., & Hausdorff, J. M. (2018). Cognitive Involvement in Balance, Gait and Dual-Tasking in Aging: A Focused Review From a Neuroscience of Aging Perspective. *Frontiers in Neurology*, *9*. <https://doi.org/10.3389/fneur.2018.00913>
- Li, M., Liu, Y., Wu, Y., Liu, S., Jia, J., & Zhang, L. (2014). Neurophysiological substrates of stroke patients with motor imagery-based brain-computer interface training. *International Journal of Neuroscience*, *124*(6), 403–415. <https://doi.org/10.3109/00207454.2013.850082>
- Liang, W., Xu, Y., Schmidt, J., Zhang, L., & Ruddy, K. L. (2020). Upregulating excitability of corticospinal pathways in stroke patients using TMS neurofeedback; A pilot study. *NeuroImage: Clinical*, *28*. <https://doi.org/10.1016/j.nicl.2020.102465>
- Lin, C.-S., Ku, H.-L., Chao, H.-T., Tu, P.-C., Li, C.-T., Cheng, C.-M., Su, T.-P., Lee, Y.-C., & Hsieh, J.-C. (2014). Neural Network of Body Representation Differs between Transsexuals and Cissexuals. *PLOS ONE*, *9*(1), e85914. <https://doi.org/10.1371/journal.pone.0085914>
- Lin, C.-T., Chiu, C.-Y., Singh, A. K., King, J.-T., Ko, L.-W., Lu, Y.-C., & Wang, Y.-K. (2019). A Wireless Multifunctional SSVEP-Based Brain–Computer Interface Assistive System. *IEEE Transactions on Cognitive and Developmental Systems*, *11*(3), 375–383. <https://doi.org/10.1109/TCDS.2018.2820153>
- Little, S., Bonaiuto, J., Barnes, G., & Bestmann, S. (2019). Human motor cortical beta bursts relate to movement planning and response errors. *PLoS Biology*, *17*(10), e3000479. <https://doi.org/10.1371/journal.pbio.3000479>

- Liu, Y., Du, L., Li, Y., Liu, H., Zhao, W., Liu, D., Zeng, J., Li, X., Fu, Y., Qiu, H., Li, X., Qiu, T., Hu, H., Meng, H., & Luo, Q. (2015). Antidepressant Effects of Electroconvulsive Therapy Correlate With Subgenual Anterior Cingulate Activity and Connectivity in Depression. *Medicine*, 94(45), e2033. <https://doi.org/10.1097/MD.0000000000002033>
- Lo, J., Chan, L., & Flynn, S. (2021). A Systematic Review of the Incidence, Prevalence, Costs, and Activity and Work Limitations of Amputation, Osteoarthritis, Rheumatoid Arthritis, Back Pain, Multiple Sclerosis, Spinal Cord Injury, Stroke, and Traumatic Brain Injury in the United States: A 2019 Update. *Archives of Physical Medicine and Rehabilitation*, 102(1), 115–131. <https://doi.org/10.1016/j.apmr.2020.04.001>
- Lopez, C. (2013). A neuroscientific account of how vestibular disorders impair bodily self-consciousness. *Frontiers in Integrative Neuroscience*, 7. <https://doi.org/10.3389/fnint.2013.00091>
- López, N. D., Monge Pereira, E., Centeno, E. J., & Miangolarra Page, J. C. (2019). Motor imagery as a complementary technique for functional recovery after stroke: A systematic review. *Topics in Stroke Rehabilitation*, 26(8), 576–587. <https://doi.org/10.1080/10749357.2019.1640000>
- López-Larraz, E., Sarasola-Sanz, A., Irastorza-Landa, N., Birbaumer, N., & Ramos-Murguialday, A. (2018). Brain-machine interfaces for rehabilitation in stroke: A review. *NeuroRehabilitation*, 43(1), 77–97. Scopus. <https://doi.org/10.3233/NRE-172394>
- López-Larraz, E., Ray, A. M., Birbaumer, N., & Ramos-Murguialday, A. (2019). Sensorimotor rhythm modulation depends on resting-state oscillations and cortex integrity in severely paralyzed stroke patients. *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, 37–40. <https://doi.org/10.1109/NER.2019.8717112>
- López-Larraz, E., Ray, A. M., Figueiredo, T. C., Bibián, C., Birbaumer, N., & Ramos-Murguialday, A. (2017). Stroke lesion location influences the decoding of movement intention from EEG. *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 3065–3068. <https://doi.org/10.1109/EMBC.2017.8037504>
- Lövdén, M., Bäckman, L., Lindenberger, U., Schaefer, S., & Schmiedek, F. (2010). A theoretical framework for the study of adult cognitive plasticity. *Psychological Bulletin*, 136(4), 659–676. <https://doi.org/10.1037/a0020080>

- Losanno, E., Mender, M., Chestek, C., Shokur, S., & Micera, S. (2023). Neurotechnologies to restore hand functions. *Nature Reviews Bioengineering*, 1(6), Article 6. <https://doi.org/10.1038/s44222-023-00054-4>
- Lotte, F., & Jeunet, C. (2015). Towards improved BCI based on human learning principles. *The 3rd International Winter Conference on Brain-Computer Interface*, 1–4. <https://doi.org/10.1109/IWW-BCI.2015.7073024>
- Lotte, F., & Jeunet, C. (2017). Online classification accuracy is a poor metric to study mental imagery-based bci user learning: An experimental demonstration and new metrics. *7th International BCI Conference*. <https://doi.org/10.3217/978-3-85125-533-1-53>
- Luengo-Fernandez, R., Violato, M., Candio, P., & Leal, J. (2020). Economic burden of stroke across Europe: A population-based cost analysis. *European Stroke Journal*, 5(1), 17–25. <https://doi.org/10.1177/2396987319883160>
- Lundquist, C. B., Nielsen, J. F., Arguissain, F. G., & Brunner, I. C. (2021). Accuracy of the Upper Limb Prediction Algorithm PREP2 Applied 2 Weeks Poststroke: A Prospective Longitudinal Study. *Neurorehabilitation and Neural Repair*, 35(1), 68–78. <https://doi.org/10.1177/1545968320971763>
- Madsen, K. H., Karabanov, A. N., Krohne, L. G., Safeldt, M. G., Tomasevic, L., & Siebner, H. R. (2019). No trace of phase: Corticomotor excitability is not tuned by phase of pericentral mu-rhythm. *Brain Stimulation*, 12(5), 1261–1270. <https://doi.org/10.1016/j.brs.2019.05.005>
- Majid, D. S. A., Lewis, C., & Aron, A. R. (2015). Training voluntary motor suppression with real-time feedback of motor evoked potentials. *Journal of Neurophysiology*, 113(9), 3446–3452. <https://doi.org/10.1152/jn.00992.2014>
- Maki, B. E., & McIlroy, W. E. (1997). The Role of Limb Movements in Maintaining Upright Stance: The “Change-in-Support” Strategy. *Physical Therapy*, 77(5), 488–507. <https://doi.org/10.1093/ptj/77.5.488>
- Malouin, F., Richards, C. L., Jackson, P. L., Lafleur, M. F., Durand, A., & Doyon, J. (2007). The Kinesthetic and Visual Imagery Questionnaire (KVIQ) for assessing motor imagery in persons with physical disabilities: A reliability and construct validity study. *Journal of Neurologic Physical Therapy: JNPT*, 31(1), 20–29. <https://doi.org/10.1097/01.npt.0000260567.24122.64>

- Marchesotti, S., Bassolino, M., Serino, A., Bleuler, H., & Blanke, O. (2016). Quantifying the role of motor imagery in brain-machine interfaces. *Scientific Reports*, 6(1), Article 1. <https://doi.org/10.1038/srep24076>
- Marins, T., Rodrigues, E. C., Bortolini, T., Melo, B., Moll, J., & Tovar-Moll, F. (2019). Structural and functional connectivity changes in response to short-term neurofeedback training with motor imagery. *NeuroImage*, 194, 283–290. <https://doi.org/10.1016/j.neuroimage.2019.03.027>
- Martel, A., Arvaneh, M., Dockree, P., & Robertson, I. (2016). *Neurophysiological correlates of mind-wandering, towards a predictive BCI-based neurofeedback* [Proceedings Paper]. Proceedings of the Sixth International Brain-Computer Interface Meeting: BCI Past, Present, and Future; Graz University of Technology. <http://dx.doi.org/10.3217/978-3-85125-467-9>
- Mashat, M. E. M., Lin, C.-T., & Zhang, D. (2019). Effects of Task Complexity on Motor Imagery-Based Brain–Computer Interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(10), 2178–2185. <https://doi.org/10.1109/TNSRE.2019.2936987>
- McConnell, A. C., Moioli, R. C., Brasil, F. L., Vallejo, M., Corne, D. W., Vargas, P. A., & Stokes, A. A. (2017). Robotic devices and brain-machine interfaces for hand rehabilitation post-stroke. *Journal of Rehabilitation Medicine*, 49(6), 449–460. Scopus. <https://doi.org/10.2340/16501977-2229>
- McElwaine, P., McCormack, J., & Harbison, J. (2016). Irish Heart Foundation/HSE national stroke audit rehabilitation units 2016. Dublin: Dublin Irish Heart Foundation.
- McGeoch, P. D., Brang, D., Huang, M., & Ramachandran, V. S. (2015). Primary somatosensory cortex hand representation dynamically modulated by motor output. *Neurocase*, 21(1), 103–105. <https://doi.org/10.1080/13554794.2013.873060>
- McGeoch, P. D., Brang, D., Song, T., Lee, R. R., Huang, M., & Ramachandran, V. S. (2011). Xenomelia: A new right parietal lobe syndrome. *Journal of Neurology, Neurosurgery & Psychiatry*, 82(12), 1314–1319. <https://doi.org/10.1136/jnnp-2011-300224>
- McWilliams, E. C., Barbey, F. M., Dyer, J. D., Islam, Md. N., McGuinness, B., Murphy, B., Nolan, H., Passmore, P., Rueda-Delgado, L. M., & Buick, A. R. (2021). *Feasibility of repeated assessment of cognitive function in older adults using a wireless, mobile, dry-EEG headset and tablet-based games*.

- Memon, S. A., Waheed, A., Başaklar, T., & Ider Ider, Y. ziya. (2018). Low-Cost Portable 4-Channel Wireless EEG Data Acquisition System for BCI Applications. *2018 Medical Technologies National Congress (TIPTEKNO)*, 1–4. <https://doi.org/10.1109/TIPTEKNO.2018.8597134>
- Mennella, R., Patron, E., & Palomba, D. (2017). Frontal alpha asymmetry neurofeedback for the reduction of negative affect and anxiety. *Behaviour Research and Therapy*, 92, 32–40. <https://doi.org/10.1016/j.brat.2017.02.002>
- Mihelj, E., Bächinger, M., Kikkert, S., Ruddy, K., & Wenderoth, N. (2021). Mental individuation of imagined finger movements can be achieved using TMS-based neurofeedback. *NeuroImage*, 242, 118463. <https://doi.org/10.1016/j.neuroimage.2021.118463>
- Mirelman, A., Herman, T., Brozgol, M., Dorfman, M., Sprecher, E., Schweiger, A., Giladi, N., & Hausdorff, J. M. (2012). Executive Function and Falls in Older Adults: New Findings from a Five-Year Prospective Study Link Fall Risk to Cognition. *PLOS ONE*, 7(6), e40297. <https://doi.org/10.1371/journal.pone.0040297>
- Miyakoshi, M. (2021, January 19). *Github, sccn/clean_rawdata Wiki*. GitHub. https://github.com/sccn/clean_rawdata
- Monge-Pereira, E., Ibañez-Pereda, J., Alguacil-Diego, I. M., Serrano, J. I., Spottorno-Rubio, M. P., & Molina-Rueda, F. (2017). Use of Electroencephalography Brain-Computer Interface Systems as a Rehabilitative Approach for Upper Limb Function After a Stroke: A Systematic Review. *PM and R*, 9(9), 918–932. Scopus. <https://doi.org/10.1016/j.pmrj.2017.04.016>
- Monteiro, K. B., Cardoso, M. dos S., Cabral, V. R. da C., Santos, A. O. B. dos, Silva, P. S. da, Castro, J. B. P. de, & Vale, R. G. de S. (2021). Effects of Motor Imagery as a Complementary Resource on the Rehabilitation of Stroke Patients: A Meta-Analysis of Randomized Trials. *Journal of Stroke and Cerebrovascular Diseases*, 30(8), 105876. <https://doi.org/10.1016/j.jstrokecerebrovasdis.2021.105876>
- Montero-Odasso, M., Verghese, J., Beauchet, O., & Hausdorff, J. M. (2012). Gait and Cognition: A Complementary Approach to Understanding Brain Function and the Risk of Falling. *Journal of the American Geriatrics Society*, 60(11), 2127–2136. <https://doi.org/10.1111/j.1532-5415.2012.04209.x>
- Moro, V., Pernigo, S., Tsakiris, M., Avesani, R., Edelstyn, N. M. J., Jenkinson, P. M., & Fotopoulou, A. (2016). Motor versus body awareness: Voxel-based lesion analysis in anosognosia for hemiplegia and

somatoparaphrenia following right hemisphere stroke. *Cortex*, 83, 62–77.
<https://doi.org/10.1016/j.cortex.2016.07.001>

Mrachacz-Kersting, N., Ibáñez, J., & Farina, D. (2021). Towards a mechanistic approach for the development of non-invasive brain-computer interfaces for motor rehabilitation. *The Journal of Physiology*, 599(9), 2361–2374. <https://doi.org/10.1113/JP281314>

Mufaddel, A., Osman, O. T., Almugaddam, F., & Jafferany, M. (2013). A Review of Body Dysmorphic Disorder and Its Presentation in Different Clinical Settings. *The Primary Care Companion for CNS Disorders*, 15(4), 0–0. <https://doi.org/10.4088/PCC.12r01464>

Müller, S. (2009). Body Integrity Identity Disorder (BIID)—Is the Amputation of Healthy Limbs Ethically Justified? *The American Journal of Bioethics*, 9(1), 36–43.
<https://doi.org/10.1080/15265160802588194>

Muir, S. W., Gopaul, K., & Montero Odasso, M. M. (2012). The role of cognitive impairment in fall risk among older adults: A systematic review and meta-analysis. *Age and Ageing*, 41(3), 299–308.
<https://doi.org/10.1093/ageing/afs012>

Mullen, T. R., Kothe, C. A. E., Chi, Y. M., Ojeda, A., Kerth, T., Makeig, S., Jung, T.-P., & Cauwenberghs, G. (2015). Real-time neuroimaging and cognitive monitoring using wearable dry EEG. *IEEE Transactions on Biomedical Engineering*, 62(11), 2553–2567.
<https://doi.org/10.1109/TBME.2015.2481482>

Mullen, T. R., Kothe, C., Chi, Y. M., Ojeda, A., Kerth, T., Makeig, S., Cauwenberghs, G., & Jung, T.-P. (2013). Real-time modeling and 3D visualization of source dynamics and connectivity using wearable EEG. *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2184–2187. <https://doi.org/10.1109/EMBC.2013.6609968>

Murphy, B., Aleni, A., Belaoucha, B., Dyer, J., & Nolan, H. (2018a). Quantifying cognitive aging and performance with at-home gamified mobile EEG. *2018 International Workshop on Pattern Recognition in Neuroimaging (PRNI)*, 1–4. <https://doi.org/10.1109/PRNI.2018.8423954>

Murphy, B., Barbey, F., Buick, A. R., Dyer, J., Farina, F., McGuinness, B., Passmore, A. P., & Whelan, R. (2019). F3-03-03: Replicating lab electrophysiology with older users in the home, using gamified dry eeg. *Alzheimer's & Dementia*, 15(7S_Part_16), P867–P867.
<https://doi.org/10.1016/j.jalz.2019.06.4606>

- Murphy, B., Buick, A. R., Dyer, J., Nolan, H., McGuinness, B., & Passmore, A. P. (2018b). P4-317: Measuring cognitive decline with home-based gamified mobile eeg. *Alzheimer's & Dementia*, 14(7S_Part_30), P1579–P1579. <https://doi.org/10.1016/j.jalz.2018.07.140>
- Nagamatsu, L. S., Boyd, L. A., Hsu, C. L., Handy, T. C., & Liu-Ambrose, T. (2013). Overall reductions in functional brain activation are associated with falls in older adults: An fMRI study. *Frontiers in Aging Neuroscience*, 5, 91. <https://doi.org/10.3389/fnagi.2013.00091>
- Nakayama, H., Jørgensen, H. S., Raaschou, H. O., & Olsen, T. S. (1994). Recovery of upper extremity function in stroke patients: The Copenhagen Stroke Study. *Archives of Physical Medicine and Rehabilitation*, 75(4), 394–398. [https://doi.org/10.1016/0003-9993\(94\)90161-9](https://doi.org/10.1016/0003-9993(94)90161-9)
- Nam, C. S., Jeon, Y., Kim, Y.-J., Lee, I., & Park, K. (2011). Movement imagery-related lateralization of event-related (de)synchronization (ERD/ERS): Motor-imagery duration effects. *Clinical Neurophysiology*, 122(3), 567–577. <https://doi.org/10.1016/j.clinph.2010.08.002>
- National Clinical Programme for Rehabilitation Medicine. (n.d.). HSE.Ie. Retrieved July 20, 2023, from <https://www.hse.ie/eng/about/who/cspd/ncps/rehabilitation-medicine/about-the-programme.html>
- Nawata, H., Ogomori, K., Tanaka, M., Nishimura, R., Urashima, H., Yano, R., Takano, K., & Kuwabara, Y. (2010). Regional cerebral blood flow changes in female to male gender identity disorder. *Psychiatry and Clinical Neurosciences*, 64(2), 157–161. <https://doi.org/10.1111/j.1440-1819.2009.02059.x>
- Neuper, C., Scherer, R., Reiner, M., & Pfurtscheller, G. (2005). Imagery of motor actions: Differential effects of kinesthetic and visual–motor mode of imagery in single-trial EEG. *Cognitive Brain Research*, 25(3), 668–677. <https://doi.org/10.1016/j.cogbrainres.2005.08.014>
- Neuper, C., Scherer, R., Wriessnegger, S., & Pfurtscheller, G. (2009). Motor imagery and action observation: Modulation of sensorimotor brain rhythms during mental control of a brain–computer interface. *Clinical Neurophysiology*, 120(2), 239–247. <https://doi.org/10.1016/j.clinph.2008.11.015>
- Nierhaus, T., Vidaurre, C., Sannelli, C., Mueller, K.-R., & Villringer, A. (2021). Immediate brain plasticity after one hour of brain–computer interface (BCI). *The Journal of Physiology*, 599(9), 2435–2451. <https://doi.org/10.1113/JP278118>
- Nojima, I., Sugata, H., Takeuchi, H., & Mima, T. (2022). Brain–Computer Interface Training Based on Brain Activity Can Induce Motor Recovery in Patients With Stroke: A Meta-Analysis. *Neurorehabilitation and Neural Repair*, 36(2), 83–96. <https://doi.org/10.1177/15459683211062895>

- Noll, S., & Kasten, E. (2014). Body Integrity Identity Disorder (BIID): How Satisfied are Successful Wannabes. *Psychology and Behavioral Sciences*, 3, 222–232. <https://doi.org/10.11648/j.pbs.20140306.17>
- Oathes, D. J., Bruce, J. M., & Nitschke, J. B. (2008). Worry facilitates corticospinal motor response to transcranial magnetic stimulation. *Depression and Anxiety*, 25(11), 969–976. <https://doi.org/10.1002/da.20445>
- Ono, T., Tomita, Y., Inose, M., Ota, T., Kimura, A., Liu, M., & Ushiba, J. (2015). Multimodal Sensory Feedback Associated with Motor Attempts Alters BOLD Responses to Paralyzed Hand Movement in Chronic Stroke Patients. *Brain Topography*, 28(2), 340–351. <https://doi.org/10.1007/s10548-014-0382-6>
- Ostgathe, A., Schnell, T., & Kasten, E. (2014). Body Integrity Identity Disorder and Gender Dysphoria: A Pilot Study to Investigate Similarities and Differences. *American Journal of Applied Psychology*, 314, 138–143. <https://doi.org/10.11648/j.ajap.20140306.14>
- Othmer, S. (2009). Neuromodulation technologies: An attempt at classification. In T. H. Budzynski, H. K. Budzynski, J. R. Evans, & A. Abarbanel (Eds.), *Introduction to quantitative EEG and neurofeedback: Advanced theory and applications* (2nd ed). Academic Press/Elsevier.
- Overman, J. J., & Carmichael, S. T. (2014). Plasticity in the Injured Brain: More than Molecules Matter. *The Neuroscientist*, 20(1), 15–28. <https://doi.org/10.1177/1073858413491146>
- Pacella, V., Ricciardi, G. K., Bonadiman, S., Verzini, E., Faraoni, F., Scandola, M., & Moro, V. (2021). The Role of White Matter Disconnection in the Symptoms Relating to the Anarchic Hand Syndrome: A Single Case Study. *Brain Sciences*, 11(5), Article 5. <https://doi.org/10.3390/brainsci11050632>
- Paciaroni, M., & Acciarresi, M. (2019). Poststroke Fatigue. *Stroke*, 50(7), 1927–1933. <https://doi.org/10.1161/STROKEAHA.119.023552>
- Park, H.-D., & Blanke, O. (2019). Coupling Inner and Outer Body for Self-Consciousness. *Trends in Cognitive Sciences*, 23(5), 377–388. <https://doi.org/10.1016/j.tics.2019.02.002>
- Pellegrini, M., Zoghi, M., & Jaberzadeh, S. (2018). The effect of transcranial magnetic stimulation test intensity on the amplitude, variability and reliability of motor evoked potentials. *Brain Research*, 1700, 190–198. <https://doi.org/10.1016/j.brainres.2018.09.002>
- Penadés, R., Catalán, R., Rubia, K., Andrés, S., Salamero, M., & Gastó, C. (2007). Impaired response inhibition in obsessive compulsive disorder. *European Psychiatry*, 22(6), 404–410. <https://doi.org/10.1016/j.eurpsy.2006.05.001>

- Perrin, F., Pernier, J., Bertrand, O., & Echallier, J. F. (1989). Spherical splines for scalp potential and current density mapping. *Electroencephalography and Clinical Neurophysiology*, 72(2), 184–187. [https://doi.org/10.1016/0013-4694\(89\)90180-6](https://doi.org/10.1016/0013-4694(89)90180-6)
- Pfefferbaum, A., Rosenbloom, M. J., Chu, W., Sassoan, S. A., Rohlfing, T., Pohl, K. M., Zahr, N. M., & Sullivan, E. V. (2014). White matter microstructural recovery with abstinence and decline with relapse in alcohol dependence interacts with normal ageing: A controlled longitudinal DTI study. *The Lancet Psychiatry*, 1(3), 202–212. [https://doi.org/10.1016/S2215-0366\(14\)70301-3](https://doi.org/10.1016/S2215-0366(14)70301-3)
- Pfurtscheller, G., Brunner, C., Schlögl, A., & Lopes da Silva, F. H. (2006). Mu rhythm (de)synchronization and EEG single-trial classification of different motor imagery tasks. *NeuroImage*, 31(1), 153–159. <https://doi.org/10.1016/j.neuroimage.2005.12.003>
- Pfurtscheller, G., & Lopes da Silva, F. H. (1999). Event-related EEG/MEG synchronization and desynchronization: Basic principles. *Clinical Neurophysiology*, 110(11), 1842–1857. [https://doi.org/10.1016/S1388-2457\(99\)00141-8](https://doi.org/10.1016/S1388-2457(99)00141-8)
- Pfurtscheller, G., & Neuper, C. (1997). Motor imagery activates primary sensorimotor area in humans. *Neuroscience Letters*, 239(2), 65–68. [https://doi.org/10.1016/S0304-3940\(97\)00889-6](https://doi.org/10.1016/S0304-3940(97)00889-6)
- Pfurtscheller, G., Neuper, C., Muller, G. R., Obermaier, B., Krausz, G., Schlogl, A., Scherer, R., Graimann, B., Keinrath, C., Skliris, D., Wortz, M., Supp, G., & Schrank, C. (2003). Graz-BCI: State of the art and clinical applications. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2), 1–4. <https://doi.org/10.1109/TNSRE.2003.814454>
- Pfurtscheller, G., Stancák, A., & Neuper, C. (1996). Post-movement beta synchronization. A correlate of an idling motor area? *Electroencephalography and Clinical Neurophysiology*, 98(4), 281–293. [https://doi.org/10.1016/0013-4694\(95\)00258-8](https://doi.org/10.1016/0013-4694(95)00258-8)
- Pichiorri, F., Morone, G., Petti, M., Toppi, J., Pisotta, I., Molinari, M., Paolucci, S., Inghilleri, M., Astolfi, L., Cincotti, F., & Mattia, D. (2015). Brain-computer interface boosts motor imagery practice during stroke recovery: BCI and Motor Imagery. *Annals of Neurology*, 77(5), 851–865. <https://doi.org/10.1002/ana.24390>
- Pieruccini-Faria, F., Lord, S. R., Toson, B., Kemmler, W., & Schoene, D. (2019). Mental Flexibility Influences the Association Between Poor Balance and Falls in Older People – A Secondary Analysis. *Frontiers in Aging Neuroscience*, 11. <https://doi.org/10.3389/fnagi.2019.00133>

- Pillette, L., Jeunet, C., Mansencal, B., N 'Kambou, R., N 'kaoua, B., & Lotte, F. (2017). PEANUT: Personalised Emotional Agent for Neurotechnology User-Training. *7th International BCI Conference*. 7th International BCI Conference, Graz, Austria. <https://doi.org/10.3217/978-3-85125-533-1-76>
- Pinter, D., Gattringer, T., Fandler-Höfler, S., Kneihsl, M., Eppinger, S., Deutschmann, H., Pichler, A., Poltrum, B., Reishofer, G., Ropele, S., Schmidt, R., & Enzinger, C. (2020). Early Progressive Changes in White Matter Integrity Are Associated with Stroke Recovery. *Translational Stroke Research*, *11*(6), 1264–1272. <https://doi.org/10.1007/s12975-020-00797-x>
- Pinto, Mauro. F., Leal, A., Lopes, F., Pais, J., Dourado, A., Sales, F., Martins, P., & Teixeira, C. A. (2022). On the clinical acceptance of black-box systems for EEG seizure prediction. *Epilepsia Open*, *7*(2), 247–259. <https://doi.org/10.1002/epi4.12597>
- Poorganji, M., Zomorodi, R., Zrenner, C., Bansal, A., Hawco, C., Hill, A. T., Hadas, I., Rajji, T. K., Chen, R., Zrenner, B., Voineskos, D., Blumberger, D. M., & Daskalakis, Z. J. (2023). Pre-Stimulus Power but Not Phase Predicts Prefrontal Cortical Excitability in TMS-EEG. *Biosensors*, *13*(2), Article 2. <https://doi.org/10.3390/bios13020220>
- Porro, C. A., Cettolo, V., Francescato, M. P., & Baraldi, P. (2000). Ipsilateral involvement of primary motor cortex during motor imagery. *European Journal of Neuroscience*, *12*(8), 3059–3063. <https://doi.org/10.1046/j.1460-9568.2000.00182.x>
- Potocanac, Z., Hoogkamer, W., Carpes, F. P., Pijnappels, M., Verschueren, S. M. P., & Duysens, J. (2014). Response inhibition during avoidance of virtual obstacles while walking. *Gait & Posture*, *39*(1), 641–644. <https://doi.org/10.1016/j.gaitpost.2013.07.125>
- Potok, W. (2022). *Acute Benefits of Transcranial Random Noise Stimulation on Sensory and Motor Processing* [Doctoral Thesis, ETH Zurich]. <https://doi.org/10.3929/ethz-b-000543956>
- Prabhakaran, S., Zarahn, E., Riley, C., Speizer, A., Chong, J. Y., Lazar, R. M., Marshall, R. S., & Krakauer, J. W. (2008). Inter-individual Variability in the Capacity for Motor Recovery After Ischemic Stroke. *Neurorehabilitation and Neural Repair*, *22*(1), 64–71. <https://doi.org/10.1177/1545968307305302>
- Prasad, G., Herman, P., Coyle, D., McDonough, S., & Crosbie, J. (2010). Applying a brain-computer interface to support motor imagery practice in people with stroke for upper limb recovery: A feasibility study. *Journal of Neuroengineering and Rehabilitation*, *7*, 60. <https://doi.org/10.1186/1743-0003-7-60>

- Price, C. (1979). *Technology Is The Answer But What Was The Question? [Speech audio recording]*. Pidgeon Digital. <https://www.pidgeondigital.com/talks/technology-is-the-answer-but-what-was-the-question-/play/>
- Prince, S. A., Adamo, K. B., Hamel, M. E., Hardt, J., Gorber, S. C., & Tremblay, M. (2008). A comparison of direct versus self-report measures for assessing physical activity in adults: A systematic review. *International Journal of Behavioral Nutrition and Physical Activity*, 5(1), 56. <https://doi.org/10.1186/1479-5868-5-56>
- Qian, X., Loo, B. R. Y., Castellanos, F. X., Liu, S., Koh, H. L., Poh, X. W. W., Krishnan, R., Fung, D., Chee, M. W., Guan, C., Lee, T.-S., Lim, C. G., & Zhou, J. (2018). Brain-computer-interface-based intervention re-normalizes brain functional network topology in children with attention deficit/hyperactivity disorder. *Translational Psychiatry*, 8(1), 149. <https://doi.org/10.1038/s41398-018-0213-8>
- Raffin, E., & Hummel, F. C. (2018). Restoring Motor Functions After Stroke: Multiple Approaches and Opportunities. *The Neuroscientist: A Review Journal Bringing Neurobiology, Neurology and Psychiatry*, 24(4), 400–416. <https://doi.org/10.1177/1073858417737486>
- Ramachandran, V. S., Brang, D., McGeoch, P. D., & Rosar, W. (2009). Sexual and Food Preference in Apotemnophilia and Anorexia: Interactions between ‘Beliefs’ and ‘Needs’ Regulated by Two-Way Connections between Body Image and Limbic Structures. *Perception*, 38(5), 775–777. <https://doi.org/10.1068/p6350>
- Ramachandran, V. S., McGeoch, P. D., & Williams, L. (2007). Can vestibular caloric stimulation be used to treat Dejerine–Roussy Syndrome? *Medical Hypotheses*, 69(3), 486–488. <https://doi.org/10.1016/j.mehy.2006.12.036>
- Ramduny, J., Bastiani, M., Huedepohl, R., Sotiropoulos, S. N., & Chechlacz, M. (2022). The association between inadequate sleep and accelerated brain ageing. *Neurobiology of Aging*, 114, 1–14. <https://doi.org/10.1016/j.neurobiolaging.2022.02.005>
- Ramos-Murguialday, A., Broetz, D., Rea, M., Läer, L., Yilmaz, Ö., Brasil, F. L., Liberati, G., Curado, M. R., Garcia-Cossio, E., Vyziotis, A., Cho, W., Agostini, M., Soares, E., Soekadar, S., Caria, A., Cohen, L. G., & Birbaumer, N. (2013). Brain-machine interface in chronic stroke rehabilitation: A controlled study: BMI in Chronic Stroke. *Annals of Neurology*, 74(1), 100–108. <https://doi.org/10.1002/ana.23879>

- Rathee, D., Cecotti, H., & Prasad, G. (2017). Single-trial effective brain connectivity patterns enhance discriminability of mental imagery tasks. *Journal of Neural Engineering*, 14(5), 056005. <https://doi.org/10.1088/1741-2552/aa785c>
- Remsik, A. B., van Kan, P. L. E., Gloe, S., Gjini, K., Williams, L., Nair, V., Caldera, K., Williams, J. C., & Prabhakaran, V. (2022). BCI-FES With Multimodal Feedback for Motor Recovery Poststroke. *Frontiers in Human Neuroscience*, 16. <https://www.frontiersin.org/articles/10.3389/fnhum.2022.725715>
- Remsik, A., Young, B., Vermilyea, R., Kiekhoefer, L., Abrams, J., Evander Elmore, S., Schultz, P., Nair, V., Edwards, D., Williams, J., & Prabhakaran, V. (2016). A review of the progression and future implications of brain-computer interface therapies for restoration of distal upper extremity motor function after stroke. *Expert Review of Medical Devices*, 13(5), 445–454. Scopus. <https://doi.org/10.1080/17434440.2016.1174572>
- Renard, Y., Lotte, F., Gibert, G., Congedo, M., Maby, E., Delannoy, V., Bertrand, O., & Lécuyer, A. (2010). OpenViBE: An Open-Source Software Platform to Design, Test, and Use Brain–Computer Interfaces in Real and Virtual Environments. *Presence: Teleoperators and Virtual Environments*, 19(1), 35–53. <https://doi.org/10.1162/pres.19.1.35>
- Rey-Mermet, A., & Gade, M. (2018). Inhibition in aging: What is preserved? What declines? A meta-analysis. *Psychonomic Bulletin & Review*, 25(5), 1695–1716. <https://doi.org/10.3758/s13423-017-1384-7>
- Rey-Mermet, A., Gade, M., & Oberauer, K. (2018). Should we stop thinking about inhibition? Searching for individual and age differences in inhibition ability. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 44(4), 501–526. <https://doi.org/10.1037/xlm0000450>
- Roc, A., Pillette, L., Mladenovic, J., Benaroch, C., N’Kaoua, B., Jeunet, C., & Lotte, F. (2021). A review of user training methods in brain computer interfaces based on mental tasks. *Journal of Neural Engineering*, 18(1), 011002. <https://doi.org/10.1088/1741-2552/abca17>
- Rocha, L. S. O., Gama, G. C. B., Rocha, R. S. B., Rocha, L. de B., Dias, C. P., Santos, L. L. S., Santos, M. C. de S., Montebelo, M. I. de L., & Teodori, R. M. (2021). Constraint Induced Movement Therapy Increases Functionality and Quality of Life after Stroke. *Journal of Stroke and Cerebrovascular Diseases*, 30(6), 105774. <https://doi.org/10.1016/j.jstrokecerebrovasdis.2021.105774>

- Romano, D., Gandola, M., Bottini, G., & Maravita, A. (2014). Arousal responses to noxious stimuli in somatoparaphrenia and anosognosia: Clues to body awareness. *Brain: A Journal of Neurology*, 137(Pt 4), 1213–1223. <https://doi.org/10.1093/brain/awu009>
- Romano, D., & Maravita, A. (2019). The dynamic nature of the sense of ownership after brain injury. Clues from asomatognosia and somatoparaphrenia. *Neuropsychologia*, 132, 107119. <https://doi.org/10.1016/j.neuropsychologia.2019.107119>
- Ros, T., Enriquez-Geppert, S., Zotev, V., Young, K. D., Wood, G., Whitfield-Gabrieli, S., Wan, F., Vuilleumier, P., Vialatte, F., Van De Ville, D., Todder, D., Surmeli, T., Sulzer, J. S., Strehl, U., Sterman, M. B., Steiner, N. J., Sorger, B., Soekadar, S. R., Sitaram, R., ... Thibault, R. T. (2020). Consensus on the reporting and experimental design of clinical and cognitive-behavioural neurofeedback studies (CRED-nf checklist). *Brain*, 143(6), 1674–1685. <https://doi.org/10.1093/brain/awaa009>
- Ros, T., J. Baars, B., Lanius, R. A., & Vuilleumier, P. (2014). Tuning pathological brain oscillations with neurofeedback: A systems neuroscience framework. *Frontiers in Human Neuroscience*, 8. <https://doi.org/10.3389/fnhum.2014.01008>
- Rossi, S., Antal, A., Bestmann, S., Bikson, M., Brewer, C., Brockmüller, J., Carpenter, L. L., Cincotta, M., Chen, R., Daskalakis, J. D., Di Lazzaro, V., Fox, M. D., George, M. S., Gilbert, D., Kimiskidis, V. K., Koch, G., Ilmoniemi, R. J., Lefaucheur, J. P., Leocani, L., ... Hallett, M. (2021). Safety and recommendations for TMS use in healthy subjects and patient populations, with updates on training, ethical and regulatory issues: Expert Guidelines. *Clinical Neurophysiology*, 132(1), 269–306. <https://doi.org/10.1016/j.clinph.2020.10.003>
- RStudio Team. (2022). *RStudio: Integrated Development Environment for R* (2022.12.0+353) [Computer software]. <http://www.rstudio.com/>
- Ruddy, K., Balsters, J., Mantini, D., Liu, Q., Kassraian-Fard, P., Enz, N., Mihelj, E., Subhash Chander, B., Soekadar, S. R., & Wenderoth, N. (2018). Neural activity related to volitional regulation of cortical excitability. *ELife*, 7, e40843. <https://doi.org/10.7554/eLife.40843>
- Ruddy, K., & Simon, C. (2023). *Optimising control of Motor Imagery-based BCIs by priming using TMS-Neurofeedback*. <https://doi.org/10.17605/OSF.IO/P24X5>
- Ryan, C. J. (2008). Out on a Limb: The Ethical Management of Body Integrity Identity Disorder. *Neuroethics*, 13. <https://doi.org/10.1007/s12152-008-9026-4>

- Rydalch, G., Bell, H. B., Ruddy, K. L., & Bolton, D. A. E. (2019). Stop-signal reaction time correlates with a compensatory balance response. *Gait & Posture*, 71, 273–278. <https://doi.org/10.1016/j.gaitpost.2019.05.015>
- Sabaté, M., González, B., & Rodríguez, M. (2004). Brain lateralization of motor imagery: Motor planning asymmetry as a cause of movement lateralization. *Neuropsychologia*, 42(8), 1041–1049. <https://doi.org/10.1016/j.neuropsychologia.2003.12.015>
- Saetta, G., Hänggi, J., Gandola, M., Zapparoli, L., Salvato, G., Berlingeri, M., Sberna, M., Paulesu, E., Bottini, G., & Brugger, P. (2020). Neural Correlates of Body Integrity Dysphoria. *Current Biology*, 30(11), 2191–2195.e3. <https://doi.org/10.1016/j.cub.2020.04.001>
- Saetta, G., Zindel-Geisseler, O., Stauffacher, F., Serra, C., Vannuscorps, G., & Brugger, P. (2021). Asomatognosia: Structured Interview and Assessment of Visuomotor Imagery. *Frontiers in Psychology*, 11. <https://doi.org/10.3389/fpsyg.2020.544544>
- Saibene, A., Caglioni, M., Corchs, S., & Gasparini, F. (2023). EEG-Based BCIs on Motor Imagery Paradigm Using Wearable Technologies: A Systematic Review. *Sensors*, 23(5), Article 5. <https://doi.org/10.3390/s23052798>
- Sala, S. D. (2009). Dr. Strangelove syndrome. *Cortex*, 45(10), 1278–1279. <https://doi.org/10.1016/j.cortex.2008.11.011>
- Salvato, G., Richter, F., Sedeño, L., Bottini, G., & Paulesu, E. (2020). Building the bodily self-awareness: Evidence for the convergence between interoceptive and exteroceptive information in a multilevel kernel density analysis study. *Human Brain Mapping*, 41(2), 401–418. <https://doi.org/10.1002/hbm.24810>
- Sampaio-Baptista, C., Neyedli, H. F., Sanders, Z.-B., Diosi, K., Havard, D., Huang, Y., Andersson, J. L. R., Lühns, M., Goebel, R., & Johansen-Berg, H. (2020). *Neurofeedback fMRI in the motor system elicits bi-directional changes in activity and white-matter structure in the healthy adult human brain* | *bioRxiv*. <https://www.biorxiv.org/content/10.1101/2020.08.06.234526v1.abstract>
- Sau Wai Tung, Cuntai Guan, Kai Keng Ang, Kok Soon Phua, Chuanchu Wang, Ling Zhao, Wei Peng Teo, & Chew, E. (2013). Motor imagery BCI for upper limb stroke rehabilitation: An evaluation of the EEG recordings using coherence analysis. *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 261–264. <https://doi.org/10.1109/EMBC.2013.6609487>

- Schaefer, M., Heinze, H.-J., & Galazky, I. (2010). Alien Hand Syndrome: Neural Correlates of Movements without Conscious Will. *PLOS ONE*, 5(12), e15010. <https://doi.org/10.1371/journal.pone.0015010>
- Schlögl, A., Keinrath, C., Zimmermann, D., Scherer, R., Leeb, R., & Pfurtscheller, G. (2007). A fully automated correction method of EOG artifacts in EEG recordings. *Clinical Neurophysiology*, 118(1), 98–104. <https://doi.org/10.1016/j.clinph.2006.09.003>
- Schoene, D., Delbaere, K., & Lord, S. R. (2017). Impaired Response Selection During Stepping Predicts Falls in Older People—A Cohort Study. *Journal of the American Medical Directors Association*, 18(8), 719–725. <https://doi.org/10.1016/j.jamda.2017.03.010>
- Schutter, D. J. L. G., Smits, F., & Klaus, J. (2023). Mind matters: A narrative review on affective state-dependency in non-invasive brain stimulation. *International Journal of Clinical and Health Psychology*, 23(3), 100378. <https://doi.org/10.1016/j.ijchp.2023.100378>
- Sedda, A., & Bottini, G. (2014). Apotemnophilia, body integrity identity disorder or xenomelia? Psychiatric and neurologic etiologies face each other. *Neuropsychiatric Disease and Treatment*, 10, 1255–1265. <https://doi.org/10.2147/NDT.S53385>
- Seghezzi, S., Giannini, G., & Zapparoli, L. (2019). Neurofunctional correlates of body-ownership and sense of agency: A meta-analytical account of self-consciousness. *Cortex*, 121, 169–178. <https://doi.org/10.1016/j.cortex.2019.08.018>
- Seiler, B. D., Newman-Norlund, R. D., & Monsma, E. V. (2017). Inter-individual neural differences in movement imagery abilities. *Psychology of Sport and Exercise*, 30, 153–163. <https://doi.org/10.1016/j.psychsport.2017.02.007>
- Shen, X., Tokoglu, F., Papademetris, X., & Constable, R. T. (2013). Groupwise whole-brain parcellation from resting-state fMRI data for network node identification. *NeuroImage*, 82, 403–415. <https://doi.org/10.1016/j.neuroimage.2013.05.081>
- Sheridan, P. L., & Hausdorff, J. M. (2007). The role of higher-level cognitive function in gait: Executive dysfunction contributes to fall risk in Alzheimer's disease. *Dementia and Geriatric Cognitive Disorders*, 24(2), 125–137. <https://doi.org/10.1159/000105126>
- Si-Mohammed, H., Petit, J., Jeunet, C., Argelaguet, F., Spindler, F., Evain, A., Roussel, N., Casiez, G., & Lecuyer, A. (2018). Towards BCI-Based Interfaces for Augmented Reality: Feasibility, Design and

Evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 26(3), 1608–1621.
<https://doi.org/10.1109/TVCG.2018.2873737>

Simon, C., Bolton, D. A. E., Kennedy, N. C., Soekadar, S. R., & Ruddy, K. L. (2021). Challenges and Opportunities for the Future of Brain-Computer Interface in Neurorehabilitation. *Frontiers in Neuroscience*, 15, 814. <https://doi.org/10.3389/fnins.2021.699428>

Simon, C., & Ruddy, K. L. (2022). A wireless, wearable Brain-Computer Interface for neurorehabilitation at home; A feasibility study. *2022 10th International Winter Conference on Brain-Computer Interface (BCI)*, 1–6. <https://doi.org/10.1109/BCI53720.2022.9734849>

Simpson, L. A., Miller, W. C., & Eng, J. J. (2011). Effect of stroke on fall rate, location and predictors: A prospective comparison of older adults with and without stroke. *PloS One*, 6(4), e19431. <https://doi.org/10.1371/journal.pone.0019431>

Slaats-Willemse, D., Swaab-Barneveld, H., de Sonnevile, L., van der Meulen, E., & Buitelaar, J. (2003). Deficient Response Inhibition as a Cognitive Endophenotype of ADHD. *Journal of the American Academy of Child & Adolescent Psychiatry*, 42(10), 1242–1248. <https://doi.org/10.1097/00004583-200310000-00016>

Smith, M.-C., Ackerley, S. J., Barber, P. A., Byblow, W. D., & Stinear, C. M. (2019). PREP2 Algorithm Predictions Are Correct at 2 Years Poststroke for Most Patients. *Neurorehabilitation and Neural Repair*, 33(8), 635–642. <https://doi.org/10.1177/1545968319860481>

Soekadar, S. R., Kohl, S. H., Mihara, M., & von Lüthmann, A. (2021). Optical brain imaging and its application to neurofeedback. *NeuroImage: Clinical*, 30, 102577. <https://doi.org/10.1016/j.nicl.2021.102577>

Soekadar, S. R., Nann, M., Crea, S., Trigili, E., Gómez, C., Opisso, E., Cohen, L. G., Birbaumer, N., & Vitiello, N. (2019). Restoration of Finger and Arm Movements Using Hybrid Brain/Neural Assistive Technology in Everyday Life Environments. In C. Guger, N. Mrachacz-Kersting, & B. Z. Allison (Eds.), *Brain-Computer Interface Research: A State-of-the-Art Summary 7* (pp. 53–61). Springer International Publishing. https://doi.org/10.1007/978-3-030-05668-1_5

Soekadar, S. R., Witkowski, M., Gómez, C., Opisso, E., Medina, J., Cortese, M., Cempini, M., Carrozza, M. C., Cohen, L. G., Birbaumer, N., & Vitiello, N. (2016). Hybrid EEG/EOG-based brain/neural hand exoskeleton restores fully independent daily living activities after quadriplegia. *Science Robotics*, 1(1), eaag3296. <https://doi.org/10.1126/scirobotics.aag3296>

- Sollfrank, T., Hart, D., Goodsell, R., Foster, J., & Tan, T. (2015). 3D visualization of movements can amplify motor cortex activation during subsequent motor imagery. *Frontiers in Human Neuroscience*, 9. <https://doi.org/10.3389/fnhum.2015.00463>
- Sollfrank, T., Ramsay, A., Perdakis, S., Williamson, J., Murray-Smith, R., Leeb, R., Millán, J. d. R., & Kübler, A. (2016). The effect of multimodal and enriched feedback on SMR-BCI performance. *Clinical Neurophysiology*, 127(1), 490–498. <https://doi.org/10.1016/j.clinph.2015.06.004>
- Song, J., Nair, V. A., Young, B. M., Walton, L. M., Nigogosyan, Z., Remsik, A., Tyler, M. E., Farrar-Edwards, D., Caldera, K. E., Sattin, J. A., Williams, J. C., & Prabhakaran, V. (2015). DTI measures track and predict motor function outcomes in stroke rehabilitation utilizing BCI technology. *Frontiers in Human Neuroscience*, 9. <https://doi.org/10.3389/fnhum.2015.00195>
- Sparto, P. J., Fuhrman, S. I., Redfern, M. S., Jennings, J. R., Perera, S., Nebes, R. D., & Furman, J. M. (2012). Postural adjustment errors reveal deficits in inhibition during lateral step initiation in older adults. *Journal of Neurophysiology*, 109(2), 415–428. <https://doi.org/10.1152/jn.00682.2012>
- Stieger, J. R., Engel, S. A., & He, B. (2021). Continuous sensorimotor rhythm based brain computer interface learning in a large population. *Scientific Data*, 8(1), Article 1. <https://doi.org/10.1038/s41597-021-00883-1>
- Stinear, C. M., Byblow, W. D., Ackerley, S. J., Smith, M.-C., Borges, V. M., & Barber, P. A. (2017a). PREP2: A biomarker-based algorithm for predicting upper limb function after stroke. *Annals of Clinical and Translational Neurology*, 4(11), 811–820. <https://doi.org/10.1002/acn3.488>
- Stinear, C. M., Byblow, W. D., Ackerley, S. J., Smith, M.-C., Borges, V. M., & Barber, P. A. (2017b). Proportional Motor Recovery After Stroke. *Stroke*, 48(3), 795–798. <https://doi.org/10.1161/STROKEAHA.116.016020>
- Stinear, C. M., Byblow, W. D., Steyvers, M., Levin, O., & Swinnen, S. P. (2006a). Kinesthetic, but not visual, motor imagery modulates corticomotor excitability. *Experimental Brain Research*, 168(1–2), 157–164. <https://doi.org/10.1007/s00221-005-0078-y>
- Stinear, C. M., Fleming, M. K., & Byblow, W. D. (2006b). Lateralization of unimanual and bimanual motor imagery. *Brain Research*, 1095(1), 139–147. <https://doi.org/10.1016/j.brainres.2006.04.008>

- Stinear, C. M., Lang, C. E., Zeiler, S., & Byblow, W. D. (2020). Advances and challenges in stroke rehabilitation. *The Lancet Neurology*, 19(4), 348–360. [https://doi.org/10.1016/S1474-4422\(19\)30415-6](https://doi.org/10.1016/S1474-4422(19)30415-6)
- Stinear, C. M., Petoe, M. A., & Byblow, W. D. (2015). Primary Motor Cortex Excitability During Recovery After Stroke: Implications for Neuromodulation. *Brain Stimulation*, 8(6), 1183–1190. <https://doi.org/10.1016/j.brs.2015.06.015>
- Swann, N. C., Cai, W., Conner, C. R., Pieters, T. A., Claffey, M. P., George, J. S., Aron, A. R., & Tandon, N. (2012). Roles for the pre-supplementary motor area and the right inferior frontal gyrus in stopping action: Electrophysiological responses and functional and structural connectivity. *NeuroImage*, 59(3), 2860–2870. <https://doi.org/10.1016/j.neuroimage.2011.09.049>
- Tariq, M., Trivailo, P. M., & Simic, M. (2018). EEG-Based BCI Control Schemes for Lower-Limb Assistive-Robots. *Frontiers in Human Neuroscience*, 12, 312. <https://doi.org/10.3389/fnhum.2018.00312>
- Tax, C. M. W., Jeurissen, B., Vos, S. B., Viergever, M. A., & Leemans, A. (2014). Recursive calibration of the fiber response function for spherical deconvolution of diffusion MRI data. *NeuroImage*, 86, 67–80. <https://doi.org/10.1016/j.neuroimage.2013.07.067>
- Then, F. S., Luck, T., Angermeyer, M. C., & Riedel-Heller, S. G. (2016). Education as protector against dementia, but what exactly do we mean by education? *Age and Ageing*, 45(4), 523–528. <https://doi.org/10.1093/ageing/afw049>
- Thibault, R. T., MacPherson, A., Lifshitz, M., Roth, R. R., & Raz, A. (2018). Neurofeedback with fMRI: A critical systematic review. *NeuroImage*, 172, 786–807. <https://doi.org/10.1016/j.neuroimage.2017.12.071>
- Thompson, M. C. (2019). Critiquing the Concept of BCI Illiteracy. *Science and Engineering Ethics*, 25(4), 1217–1233. <https://doi.org/10.1007/s11948-018-0061-1>
- Tournier, J.-D., Calamante, F., & Connelly, A. (2007). Robust determination of the fibre orientation distribution in diffusion MRI: non-negativity constrained super-resolved spherical deconvolution. *NeuroImage*, 35(4), 1459–1472. <https://doi.org/10.1016/j.neuroimage.2007.02.016>
- Trambaiolli, L. R., Dean, P. J. A., Cravo, A. M., Sterr, A., & Sato, J. R. (2019). On-task theta power is correlated to motor imagery performance. *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 3937–3942. <https://doi.org/10.1109/SMC.2019.8913980>

- Tsakiris, M. (2010). My body in the brain: A neurocognitive model of body-ownership. *Neuropsychologia*, 48(3), 703–712. <https://doi.org/10.1016/j.neuropsychologia.2009.09.034>
- Turbyne, C., de Koning, P., Zantvoord, J., & Denys, D. (2021). Body integrity identity disorder using augmented reality: A symptom reduction study. *BMJ Case Reports*, 14(1). <https://doi.org/10.1136/bcr-2020-238554>
- Umarova, R. M., Schumacher, L. V., Schmidt, C. S. M., Martin, M., Egger, K., Urbach, H., Hennig, J., Klöppel, S., & Kaller, C. P. (2021). Interaction between cognitive reserve and age moderates effect of lesion load on stroke outcome. *Scientific Reports*, 11(1), 4478. <https://doi.org/10.1038/s41598-021-83927-1>
- United Nations, Department of Economic and Social Affairs, & Population Division. (2017). *World population ageing, 2017 highlights*.
- van der Kolk, B. A., Hodgdon, H., Gapen, M., Musicaro, R., Suvak, M. K., Hamlin, E., & Spinazzola, J. (2016). A Randomized Controlled Study of Neurofeedback for Chronic PTSD. *PLOS ONE*, 11(12), e0166752. <https://doi.org/10.1371/journal.pone.0166752>
- van der Vliet, R., Selles, R. W., Andrinopoulou, E.-R., Nijland, R., Ribbers, G. M., Frens, M. A., Meskers, C., & Kwakkel, G. (2020). Predicting Upper Limb Motor Impairment Recovery after Stroke: A Mixture Model. *Annals of Neurology*, 87(3), 383–393. <https://doi.org/10.1002/ana.25679>
- van Dijk, M. T., Wingen, G. A. van, Lammeren, A. van, Blom, R. M., Kwaasteniet, B. P. de, Scholte, H. S., & Denys, D. (2013). Neural Basis of Limb Ownership in Individuals with Body Integrity Identity Disorder. *PLOS ONE*, 8(8), e72212. <https://doi.org/10.1371/journal.pone.0072212>
- Varkuti, B., Guan, C., Pan, Y., Phua, K. S., Ang, K. K., Kuah, C. W. K., Chua, K., Ang, B. T., Birbaumer, N., & Sitaram, R. (2013). Resting State Changes in Functional Connectivity Correlate With Movement Recovery for BCI and Robot-Assisted Upper-Extremity Training After Stroke. *Neurorehabilitation and Neural Repair*, 27(1), 53–62. <https://doi.org/10.1177/1545968312445910>
- Veale, J. F. (2014). Edinburgh Handedness Inventory – Short Form: A revised version based on confirmatory factor analysis. *Laterality*, 19(2), 164–177. <https://doi.org/10.1080/1357650X.2013.783045>
- Veerbeek, J. M., Wegen, E. van, Peppen, R. van, Wees, P. J. van der, Hendriks, E., Rietberg, M., & Kwakkel, G. (2014). What Is the Evidence for Physical Therapy Poststroke? A Systematic Review and Meta-Analysis. *PLOS ONE*, 9(2), e87987. <https://doi.org/10.1371/journal.pone.0087987>

- Veraart, J., Sijbers, J., Sunaert, S., Leemans, A., & Jeurissen, B. (2013). Weighted linear least squares estimation of diffusion MRI parameters: Strengths, limitations, and pitfalls. *NeuroImage*, *81*, 335–346. <https://doi.org/10.1016/j.neuroimage.2013.05.028>
- Verhaeghen, P. (2011). Aging and Executive Control: Reports of a Demise Greatly Exaggerated. *Current Directions in Psychological Science*, *20*(3), 174–180. <https://doi.org/10.1177/0963721411408772>
- Vidaurre, C., & Blankertz, B. (2010). Towards a Cure for BCI Illiteracy. *Brain Topography*, *23*(2), 194–198. <https://doi.org/10.1007/s10548-009-0121-6>
- Virani, S. S., Alonso, A., Benjamin, E. J., Bittencourt, M. S., Callaway, C. W., Carson, A. P., Chamberlain, A. M., Chang, A. R., Cheng, S., Delling, F. N., Djousse, L., Elkind, M. S. V., Ferguson, J. F., Fornage, M., Khan, S. S., Kissela, B. M., Knutson, K. L., Kwan, T. W., Lackland, D. T., ... Turner, M.B. (2020). Heart Disease and Stroke Statistics—2020 Update: A Report From the American Heart Association. *Circulation*, *141*(9), e139–e596. <https://doi.org/10.1161/CIR.0000000000000757>
- Vos, T., Lim, S. S., Abbafati, C., Abbas, K. M., Abbasi, M., Abbasifard, M., Abbasi-Kangevari, M., Abbastabar, H., Abd-Allah, F., Abdelalim, A., Abdollahi, M., Abdollahpour, I., Abolhassani, H., Aboyans, V., Abrams, E. M., Abreu, L. G., Abrigo, M. R. M., Abu-Raddad, L. J., Abushouk, A. I., ... Murray, C. J. L. (2020). Global burden of 369 diseases and injuries in 204 countries and territories, 1990–2019: A systematic analysis for the Global Burden of Disease Study 2019. *The Lancet*, *396*(10258), 1204–1222. [https://doi.org/10.1016/S0140-6736\(20\)30925-9](https://doi.org/10.1016/S0140-6736(20)30925-9)
- Vuckovic, A., & Osuagwu, B. A. (2013). Using a motor imagery questionnaire to estimate the performance of a Brain–Computer Interface based on object oriented motor imagery. *Clinical Neurophysiology*, *124*(8), 1586–1595. <https://doi.org/10.1016/j.clinph.2013.02.016>
- Wagner, J., Wessel, J. R., Ghahremani, A., & Aron, A. R. (2018). Establishing a Right Frontal Beta Signature for Stopping Action in Scalp EEG: Implications for Testing Inhibitory Control in Other Task Contexts. *Journal of Cognitive Neuroscience*, *30*(1), 107–118. https://doi.org/10.1162/jocn_a_01183
- Wainwright, P. E., Leatherdale, S. T., & Dubin, J. A. (2007). Advantages of mixed effects models over traditional ANOVA models in developmental studies: A worked example in a mouse model of fetal alcohol syndrome. *Developmental Psychobiology*, *49*(7), 664–674. <https://doi.org/10.1002/dev.20245>
- Weijs, M. L., Ho, J. T., Roel Lesur, M., & Lenggenhager, B. (2022). Is this my foot? Experimentally induced disownership in individuals with body integrity dysphoria. *Consciousness and Cognition*, *106*, 103432. <https://doi.org/10.1016/j.concog.2022.103432>

- Whelan, B. J., & Savva, G. M. (2013). Design and Methodology of The Irish Longitudinal Study on Ageing. *Journal of the American Geriatrics Society*, 61(s2), S265–S268. <https://doi.org/10.1111/jgs.12199>
- Whelan, R., Conrod, P. J., Poline, J.-B., Lourdasamy, A., Banaschewski, T., Barker, G. J., Bellgrove, M. A., Büchel, C., Byrne, M., Cummins, T. D. R., Fauth-Bühler, M., Flor, H., Gallinat, J., Heinz, A., Ittermann, B., Mann, K., Martinot, J.-L., Lalor, E. C., Lathrop, M., ... the IMAGEN Consortium. (2012). Adolescent impulsivity phenotypes characterized by distinct brain networks. *Nature Neuroscience*, 15(6), 920–925. <https://doi.org/10.1038/nn.3092>
- White, A. (2014). Body Integrity Identity Disorder Beyond Amputation: Consent and Liberty. *HEC Forum*, 26(3), 225–236. <https://doi.org/10.1007/s10730-014-9246-4>
- Wierzgała, P., Zapła, D., Wojcik, G. M., & Masiak, J. (2018). Most Popular Signal Processing Methods in Motor-Imagery BCI: A Review and Meta-Analysis. *Frontiers in Neuroinformatics*, 12. <https://doi.org/10.3389/fninf.2018.00078>
- Winstein, C. J., Stein, J., Arena, R., Bates, B., Cherney, L. R., Cramer, S. C., Deruyter, F., Eng, J. J., Fisher, B., Harvey, R. L., Lang, C. E., MacKay-Lyons, M., Ottenbacher, K. J., Pugh, S., Reeves, M. J., Richards, L. G., Stiers, W., & Zorowitz, R. D. (2016). Guidelines for Adult Stroke Rehabilitation and Recovery. *Stroke*, 47(6), e98–e169. <https://doi.org/10.1161/STR.0000000000000098>
- Winters, C., van Wegen, E. E. H., Daffertshofer, A., & Kwakkel, G. (2015). Generalizability of the Proportional Recovery Model for the Upper Extremity After an Ischemic Stroke. *Neurorehabilitation and Neural Repair*, 29(7), 614–622. <https://doi.org/10.1177/1545968314562115>
- Wirth, C., Dockree, P. M., Harty, S., Lacey, E., & Arvaneh, M. (2019). Towards error categorisation in BCI: Single-trial EEG classification between different errors. *Journal of Neural Engineering*, 17(1), 016008. <https://doi.org/10.1088/1741-2552/ab53fe>
- Wischnewski, M., Haigh, Z. J., Shirinpour, S., Alekseichuk, I., & Opitz, A. (2022). The phase of sensorimotor mu and beta oscillations has the opposite effect on corticospinal excitability. *Brain Stimulation: Basic, Translational, and Clinical Research in Neuromodulation*, 15(5), 1093–1100. <https://doi.org/10.1016/j.brs.2022.08.005>
- World Health Organisation. (2018). *ICD-11—Mortality and Morbidity Statistics*. <https://icd.who.int/browse11/l-m/en#/http%3a%2f%2fid.who.int%2fcd%2fentfity%2f256572629>

- Wu, Q., Yue, Z., Ge, Y., Ma, D., Yin, H., Zhao, H., Liu, G., Wang, J., Dou, W., & Pan, Y. (2020). Brain Functional Networks Study of Subacute Stroke Patients With Upper Limb Dysfunction After Comprehensive Rehabilitation Including BCI Training. *Frontiers in Neurology*, 10. <https://doi.org/10.3389/fneur.2019.01419>
- Xie, Y., & Oniga, S. (2020). A Review of Processing Methods and Classification Algorithm for EEG Signal. *Carpathian Journal of Electronic and Computer Engineering*, 13(1), 23–29. <https://doi.org/10.2478/cjece-2020-0004>
- Xing, Y., & Bai, Y. (2020). A Review of Exercise-Induced Neuroplasticity in Ischemic Stroke: Pathology and Mechanisms. *Molecular Neurobiology*, 57(10), 4218–4231. <https://doi.org/10.1007/s12035-020-02021-1>
- Yang, J., Zaitlen, N. A., Goddard, M. E., Visscher, P. M., & Price, A. L. (2014). Advantages and pitfalls in the application of mixed-model association methods. *Nature Genetics*, 46(2), Article 2. <https://doi.org/10.1038/ng.2876>
- Yu, Z., Guindani, M., Grieco, S. F., Chen, L., Holmes, T. C., & Xu, X. (2022). Beyond t test and ANOVA: Applications of mixed-effects models for more rigorous statistical analysis in neuroscience research. *Neuron*, 110(1), 21–35. <https://doi.org/10.1016/j.neuron.2021.10.030>
- Zimmermann, R., Marchal-Crespo, L., Lambercy, O., Fluet, M.-C., Riener, R., Wolf, M., & Gassert, R. (2011). Towards a BCI for sensorimotor training: Initial results from simultaneous fNIRS and biosignal recordings. *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 6339–6343. <https://doi.org/10.1109/IEMBS.2011.6091565>

10. Appendices

10.1 Optimising BCI Control

10.1.1 MEP Model Assumptions

Basic Model

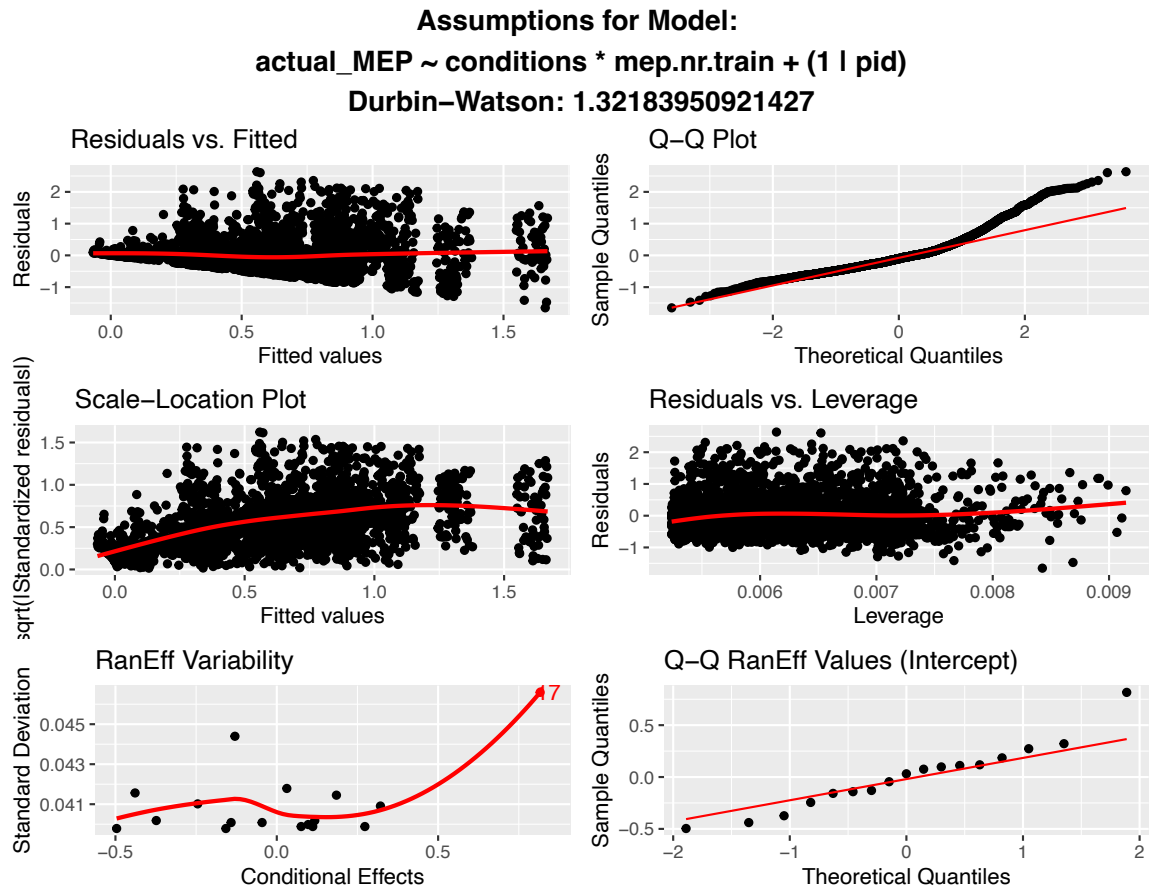


Figure 10.1. Assumptions Basic MEP Model.

Model Including Session and EEG Session Variable

Assumptions for Model:

actual_MEP ~ conditions * mep.nr.train + sess + eeg.condition + (1 | pid)

Durbin-Watson: 1.34787907610961

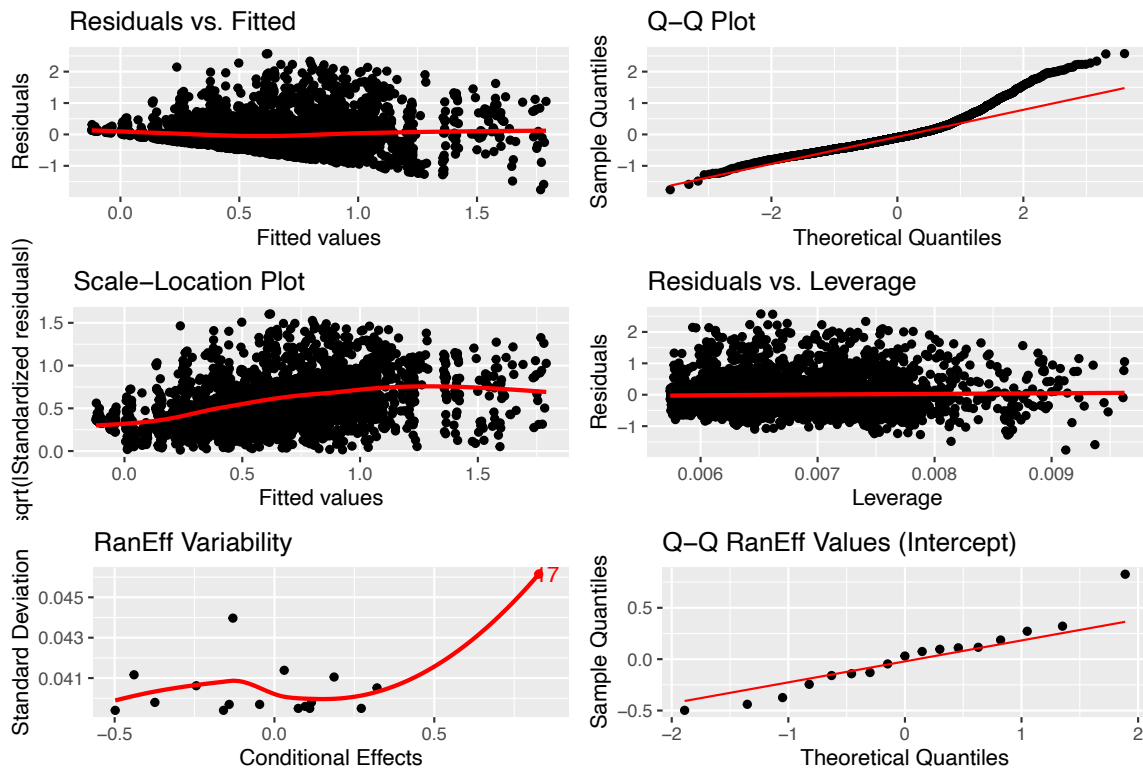


Figure 10.2. Assumptions MEP Model with Session Variable.

Model Including BrMS Variable

Assumptions for Model:

ctual_MEP ~ conditions * mep.nr.train + ztrans.rms + ztrans.rms:conditions + (1 | pic

Durbin-Watson: 1.3657659394817

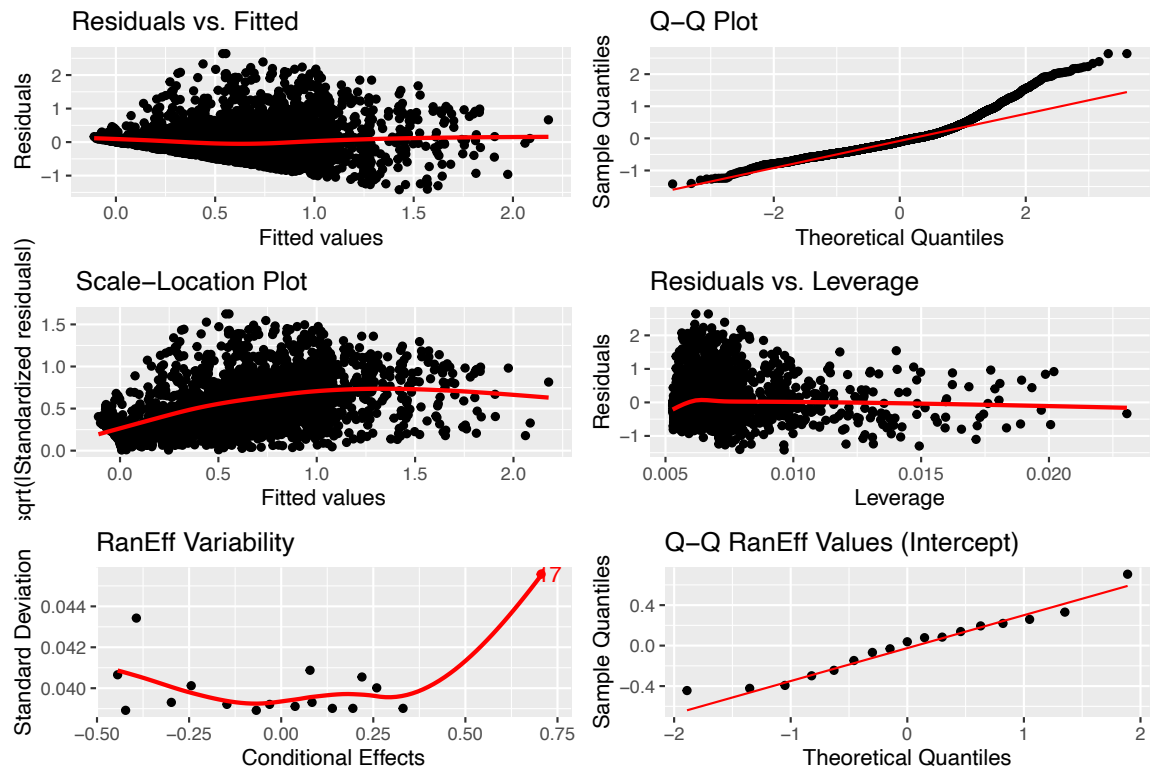


Figure 10.3. Assumptions MEP Model with BrMS.

Model Predicting BrMS

Assumptions for Model:

$\text{ztrans.rms} \sim \text{conditions} * \text{mep.nr.train} + \text{actual\_MEP} + (1 | \text{pid})$

Durbin-Watson: 1.1939222088394

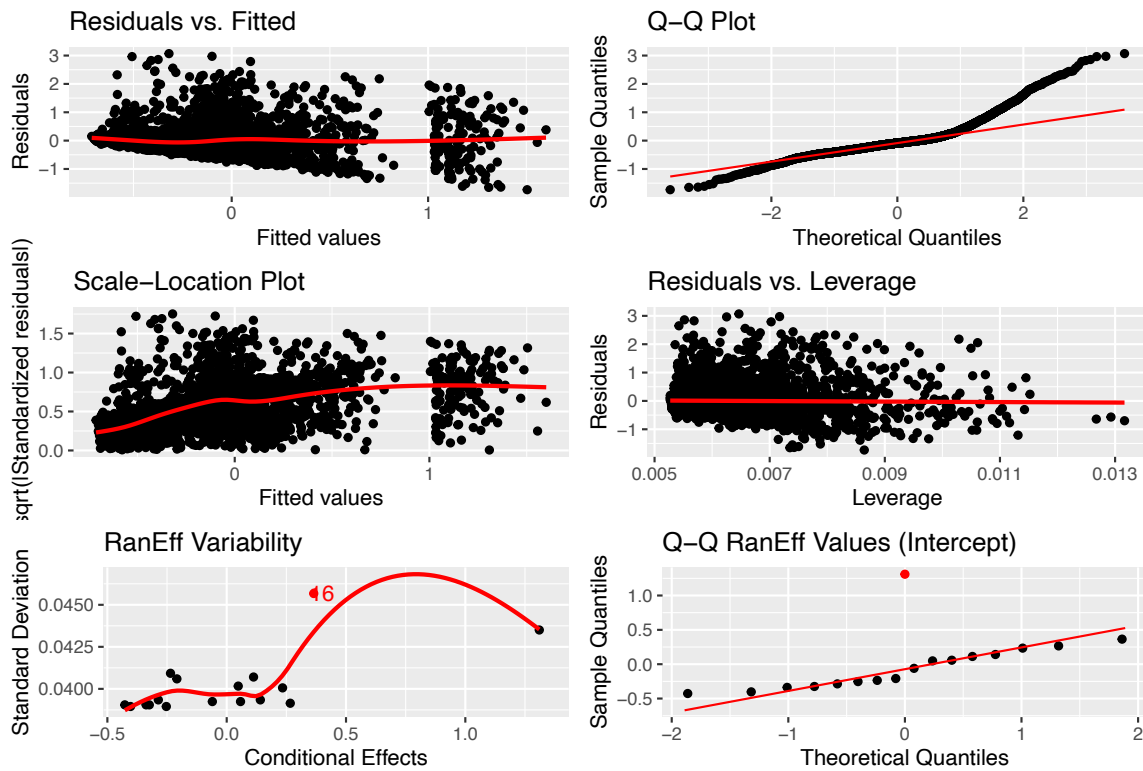


Figure 10.4. Assumptions BrMS Model.

Model Using Dynamic Thresholding MEPs

Assumptions for Model:

Dynamic.Standardised.MEP ~ conditions * mep.nr.train * ztrans.rms + (1 | pid)

Durbin–Watson: 1.67390713593237

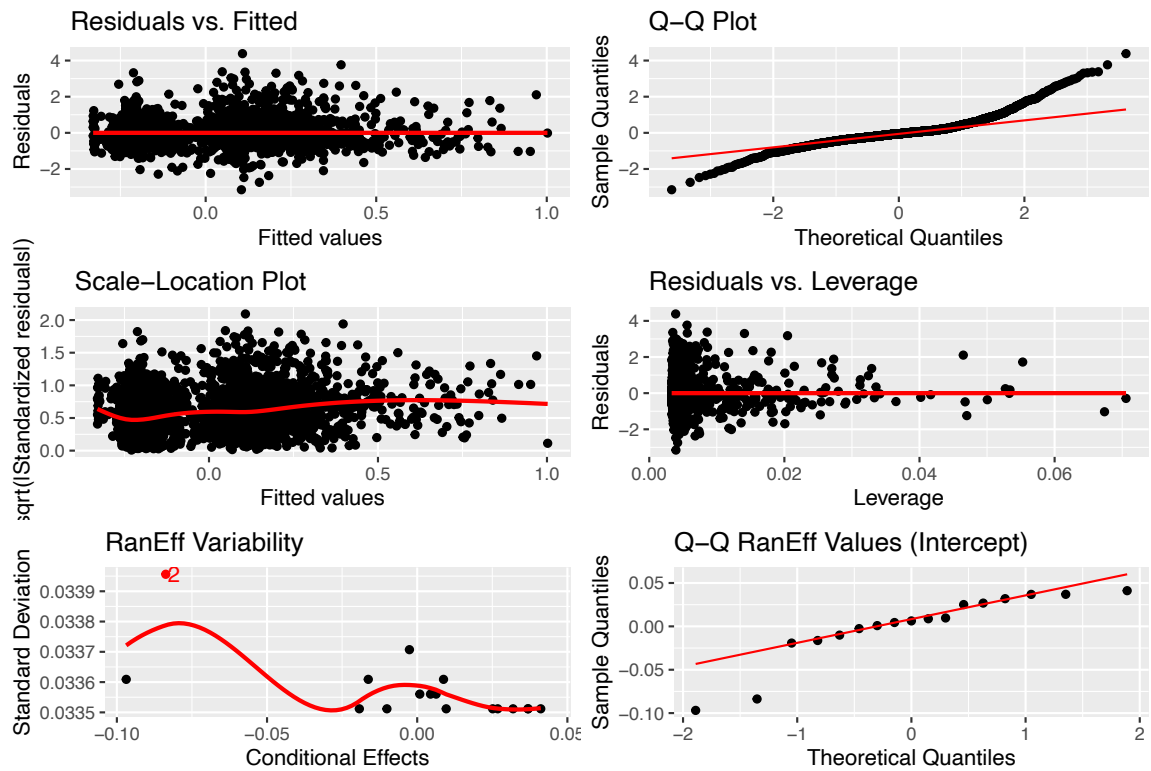


Figure 10.5. Assumptions Basic CA Model.

Model Using KVIQ Scores

Assumptions for Model:

actual_MEP ~ conditions * mep.nr.train * kviq.total + eeg.condition + sess + (1 | pid)

Durbin–Watson: 1.35212355413314

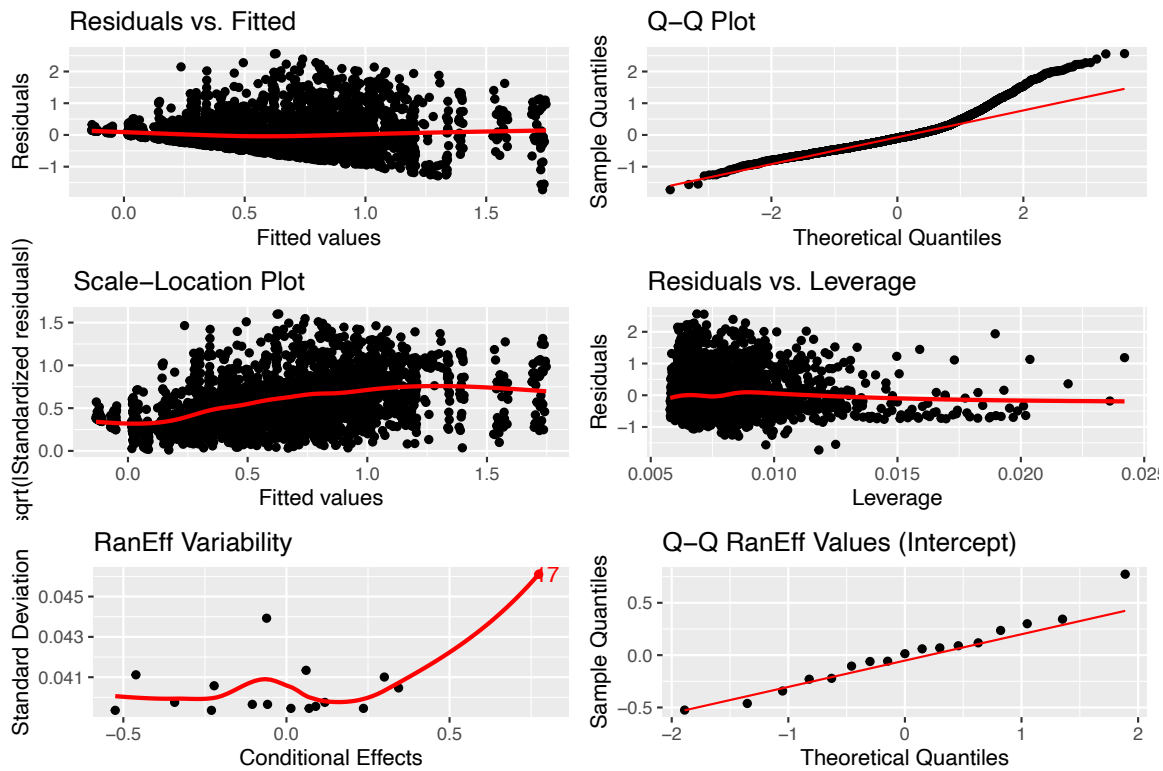


Figure 10.6. Assumptions MEP Model Using KVIQ and Session variables.

CA Model Assumptions

Base Model

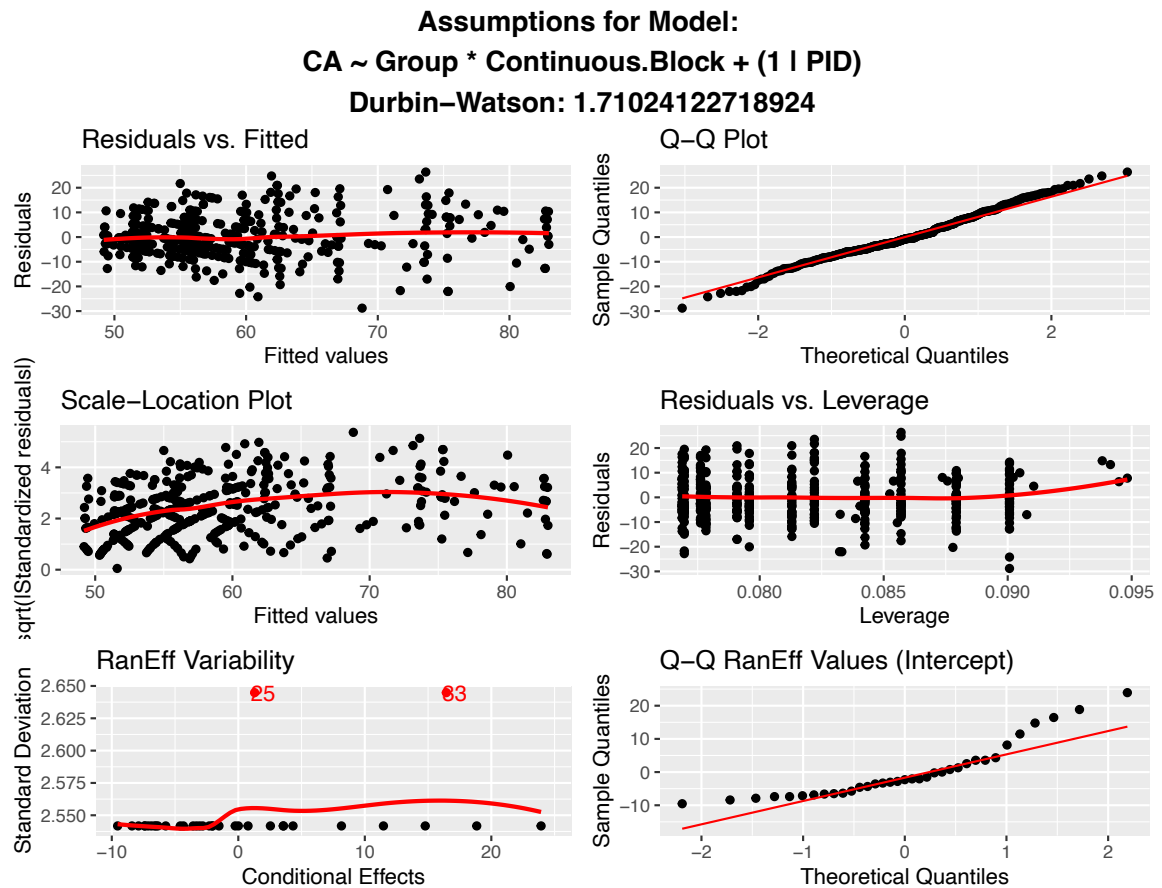


Figure 10.7. Assumptions Basic CA Model.

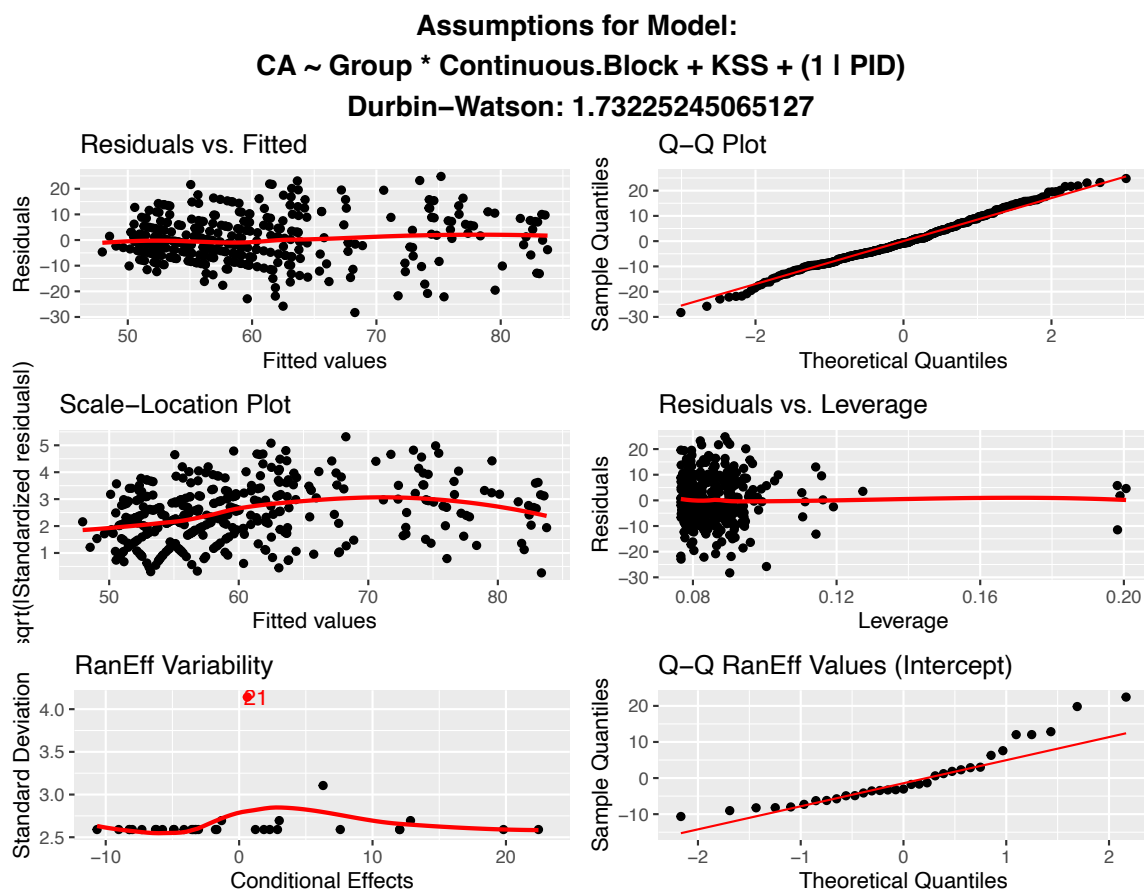


Figure 10.8. Assumptions CA Model Using KSS.

Model using KSS full interaction

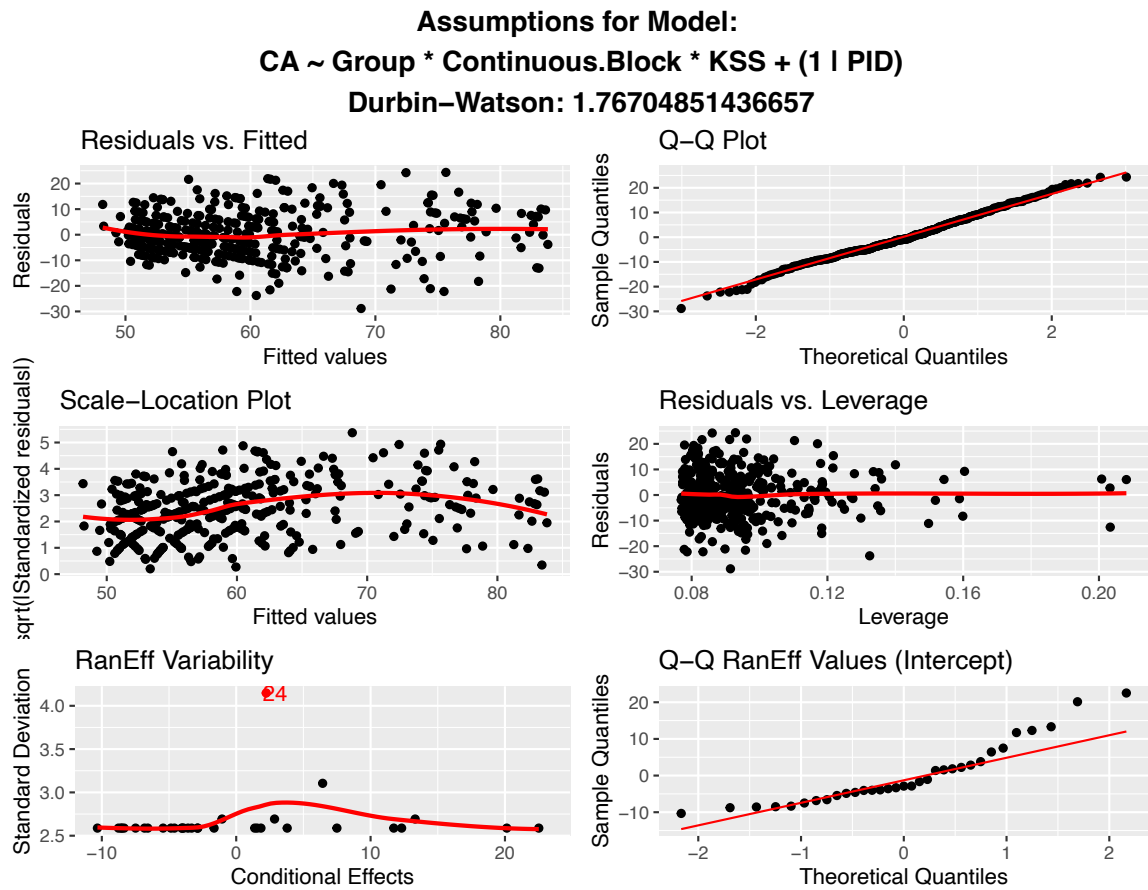


Figure 10.9. Assumptions Using KSS Full Interaction.

Assumptions for the stability and distinctiveness models

Stability of Rest, basic model

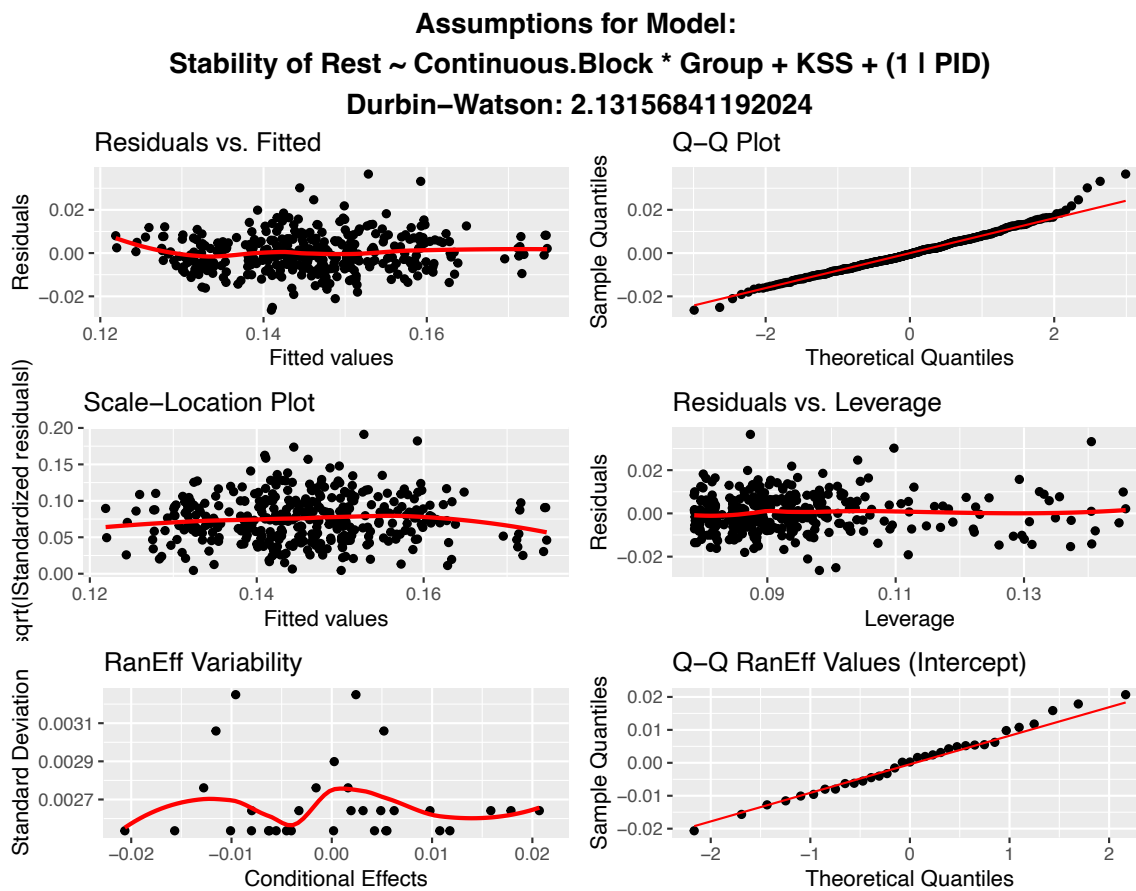


Figure 10.10. Assumptions Rest Model Using KSS.

Distinctiveness of Up vs Down, first distribution basic model

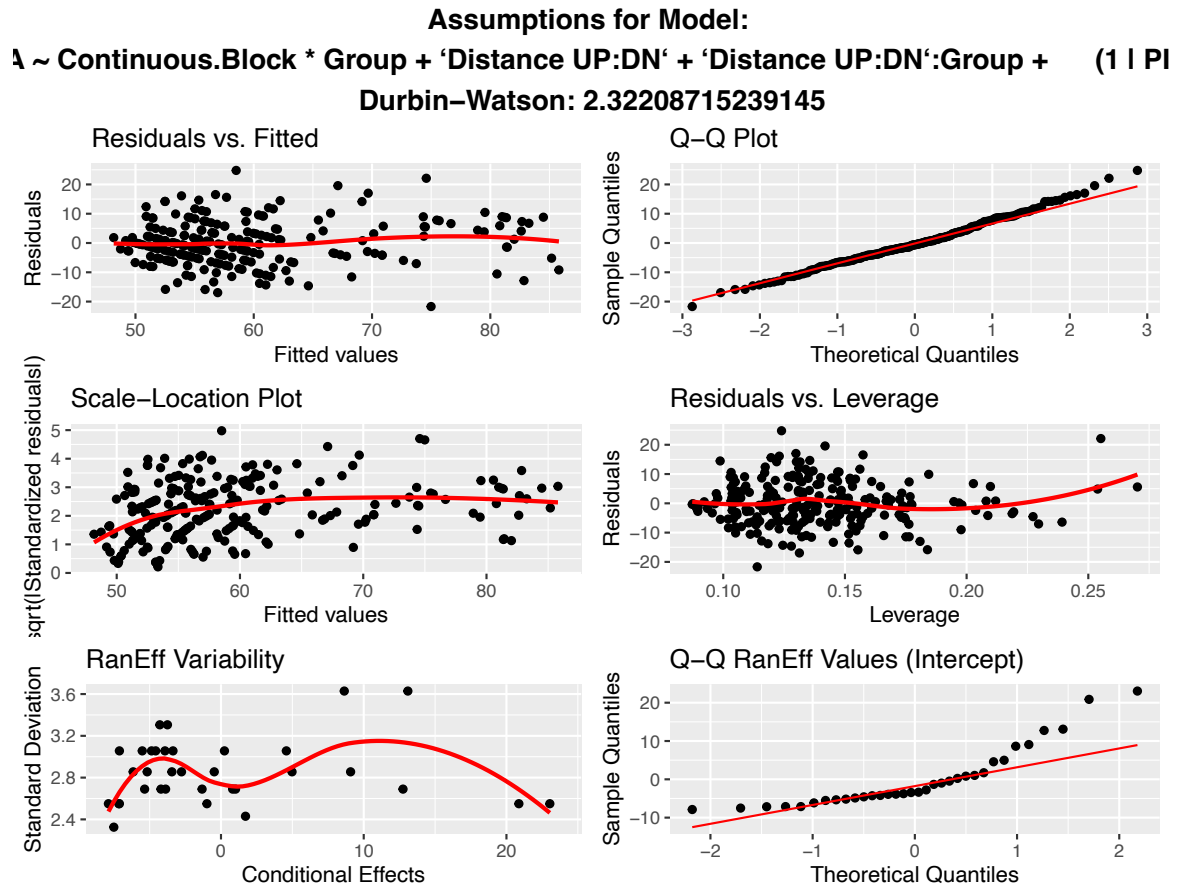


Figure 10.11. Assumptions Distinctiveness Model

Powerbands Supplementary Results

Category 1 Results

Results for CA, Low Alpha

Table 10.1

CA Model Low Alpha

	Beta	t.Value	p.Value
(Intercept)	150.54	6.87	0.001 >
Group (TMS)	-127.58	-3.96	0.001 >
Time	-6.75	-4.54	0.001 >
Low Alpha	-36.07	-4.09	0.001 >
Group (TMS):Time	10.2	4.78	0.001 >
Group (TMS):Low Alpha	48.13	3.69	0.001 >
Time:Low Alpha	2.7	4.46	0.001 >
Group (TMS):Time:Low Alpha	-3.9	-4.48	0.001 >

Association Between Relative Power, Low Alpha 8–11 Hz Power And CA Split by Session, Control and Group

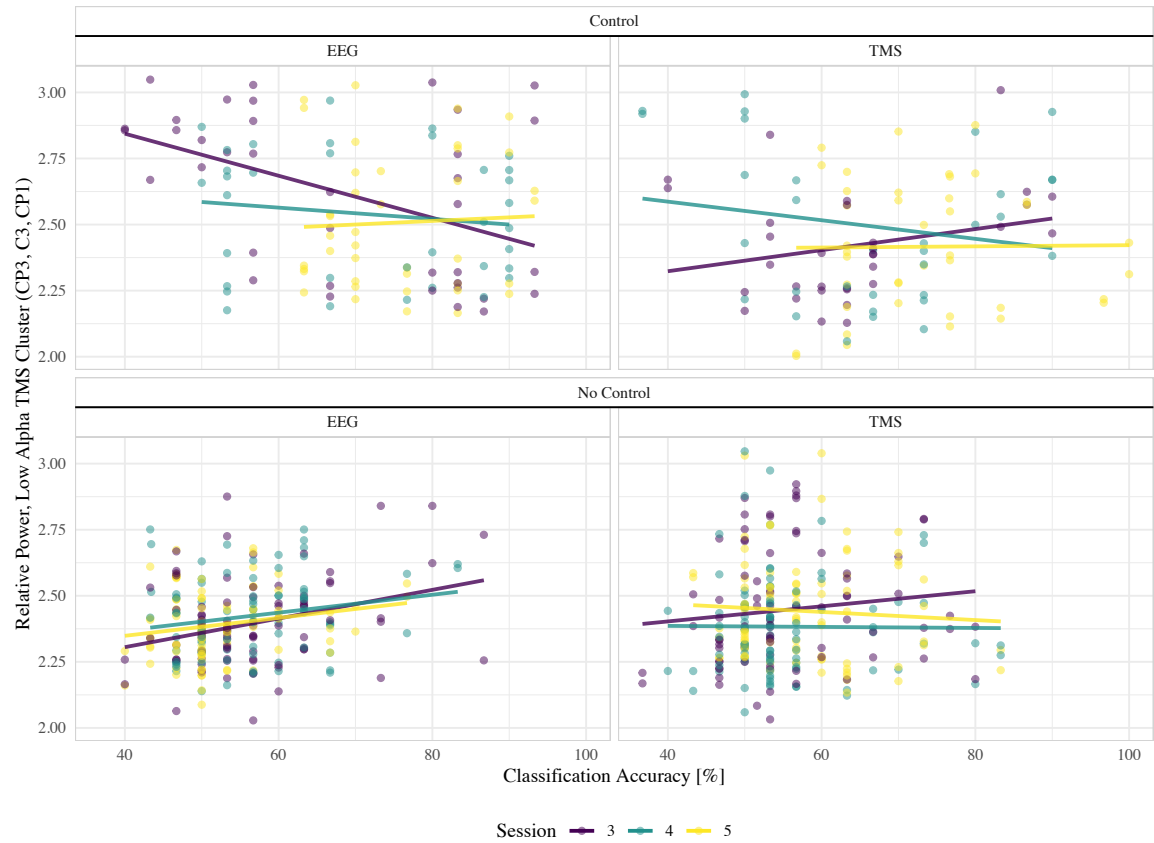


Figure 10.12. Relative Power Low Alpha Dotplot

Relative Power, TMS vs EEG Group, Comparison of Third Session vs Last Session

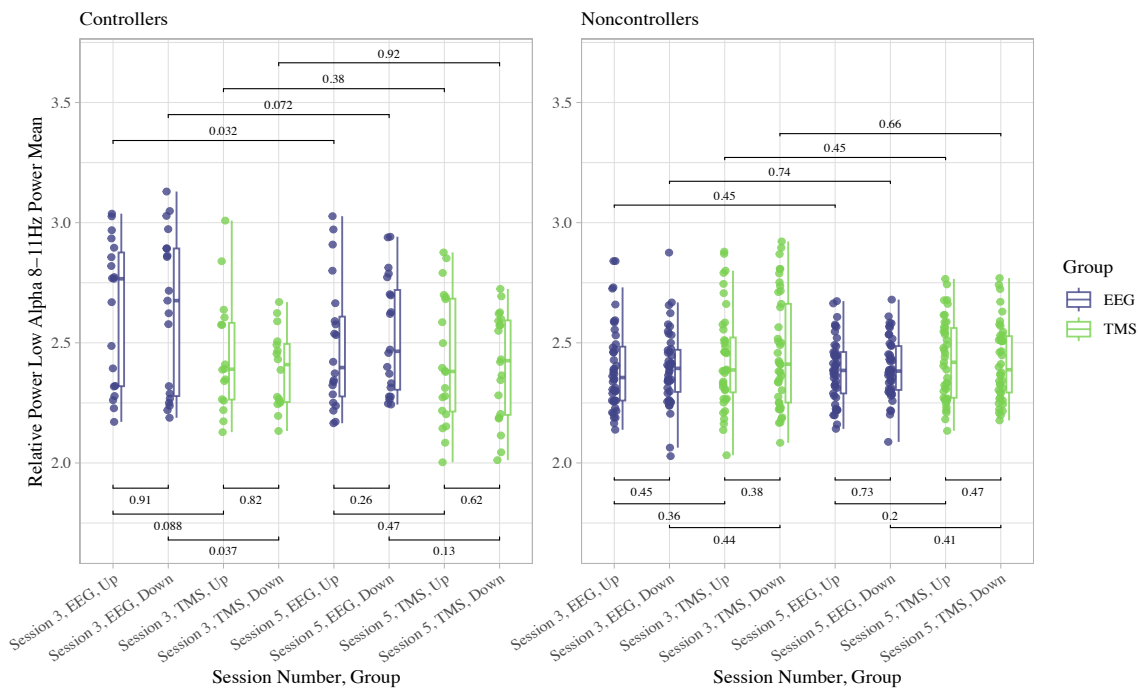


Figure 10.13. Relative Power Low Alpha Boxplot

Results for CA, Low Gamma

Table 10.2*CA Model Low Gamma*

	Beta	t.Value	p.Value
(Intercept)	-81.81	-1.39	0.1600
Group (TMS)	187.83	2.40	0.0170
Time	10.42	2.68	0.0075
Low Gamma	68.75	2.45	0.0150
Group (TMS):Time	-14.42	-2.82	0.0049
Group (TMS):Low Gamma	-94.14	-2.53	0.0120
Time:Low Gamma	-5.05	-2.72	0.0067
Group (TMS):Time:Low Gamma	7.18	2.95	0.0033

**Figure 10.14.** Relative Power Low Gamma Dotplot

Relative Power, TMS vs EEG Group, Comparison of Third Session vs Last Session

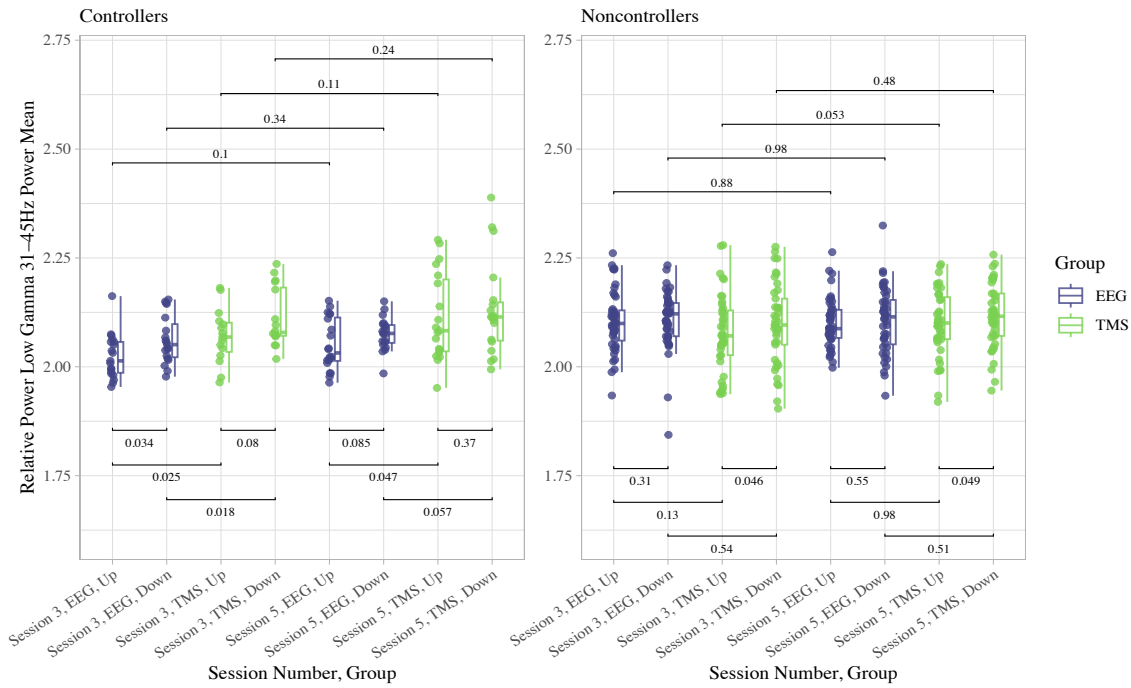


Figure 10.15. Relative Power Low Gamma Boxplot

Category 2 Results

Results for CA, High Delta

Table 10.3

CA Model High Delta

	CA Beta	CA t- Value	CA p- Value	Up Beta	Up t- Value	Up p- Value	Down Beta	Down t- Value	Down p- Value
(Intercept)	16.28	0.40	0.69	18.23	1.04	0.3	-12.01	-0.7	0.49
Group (TMS)	-50.31	-0.87	0.39	-89.83	-3.6	0.001 >	72.46	2.94	0.0034
Time	3.20	1.17	0.24	-1.65	-1.4	0.16	2.6	2.24	0.026
High Delta	19.41	1.13	0.26	-3.64	-0.49	0.62	8.89	1.22	0.22
Group (TMS):Time	2.77	0.72	0.47	7.46	4.49	0.001 >	-6.45	-3.93	0.001 >
Group (TMS):High Delta	17.24	0.70	0.48	36.32	3.44	0.001 >	-30.11	-2.89	0.004
Time:High Delta	-1.42	-1.23	0.22	0.69	1.38	0.17	-1.11	-2.26	0.024
Group (TMS):Time:High Delta	-0.90	-0.55	0.58	-3.04	-4.32	0.001 >	2.69	3.88	0.001 >

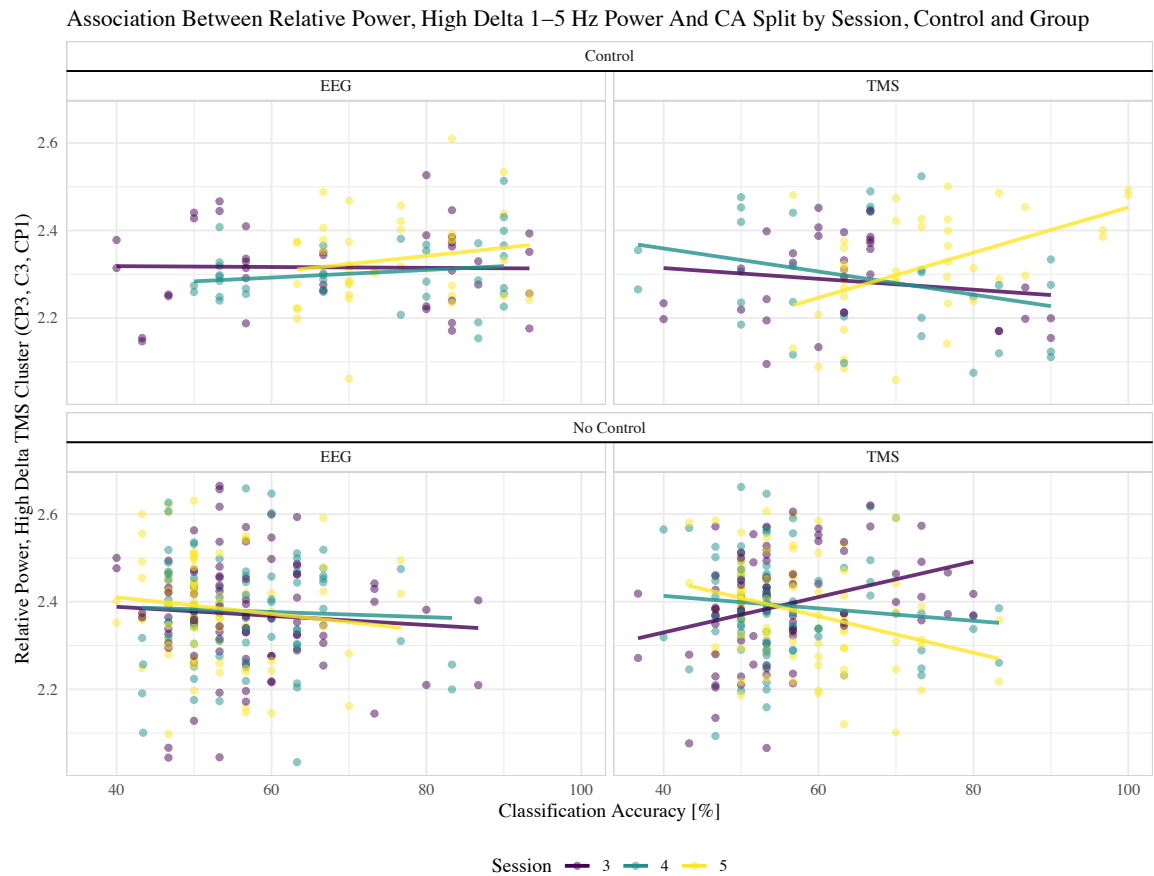


Figure 10.16. Relative Power High Delta Dotplot

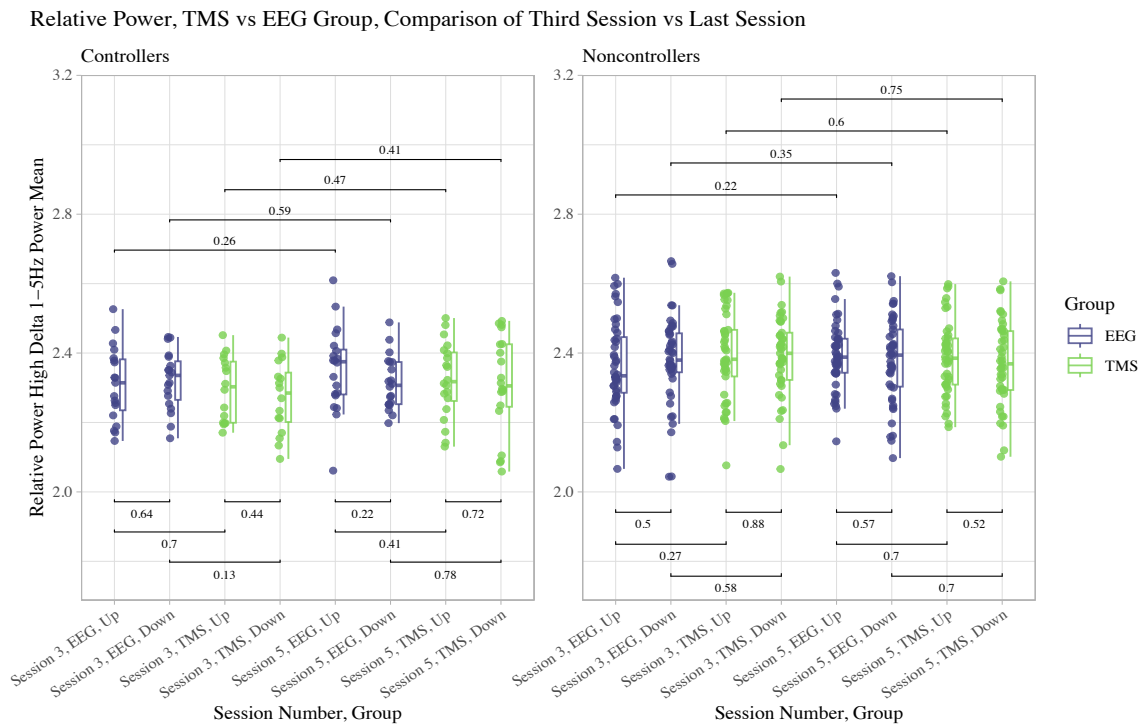


Figure 10.17. Relative Power High Delta Boxplot

Results for CA, Low Beta

Table 10.4

CA Model Low Beta

	CA Beta	CA t-Value	CA p-Value	Up Beta	Up t-Value	Up p-Value	Down Beta	Down t-Value	Down p-Value
(Intercept)	59.81	1.19	0.24	-45.91	-2.14	0.032	61.85	2.93	0.0035
Group (TMS)	16.42	0.24	0.81	151.71	5.2	0.001 >	-143.56	-5	0.001 >
Time	0.0089	0.0028	1.00	3.92	2.86	0.0044	-3.91	-2.9	0.0038
Low Beta	0.98	0.045	0.96	24.23	2.59	0.0099	-23.05	-2.5	0.013
Group (TMS):Time	-0.71	-0.16	0.87	-10.19	-5.36	0.001 >	9.76	5.22	0.001 >
Group (TMS):Low Beta	-11.4	-0.38	0.70	-68.29	-5.35	0.001 >	63.51	5.05	0.001 >
Time:Low Beta	-0.068	-0.048	0.96	-1.72	-2.87	0.0043	1.7	2.88	0.0042
Group (TMS):Time:Low Beta	0.6	0.31	0.76	4.6	5.51	0.001 >	-4.32	-5.27	0.001 >

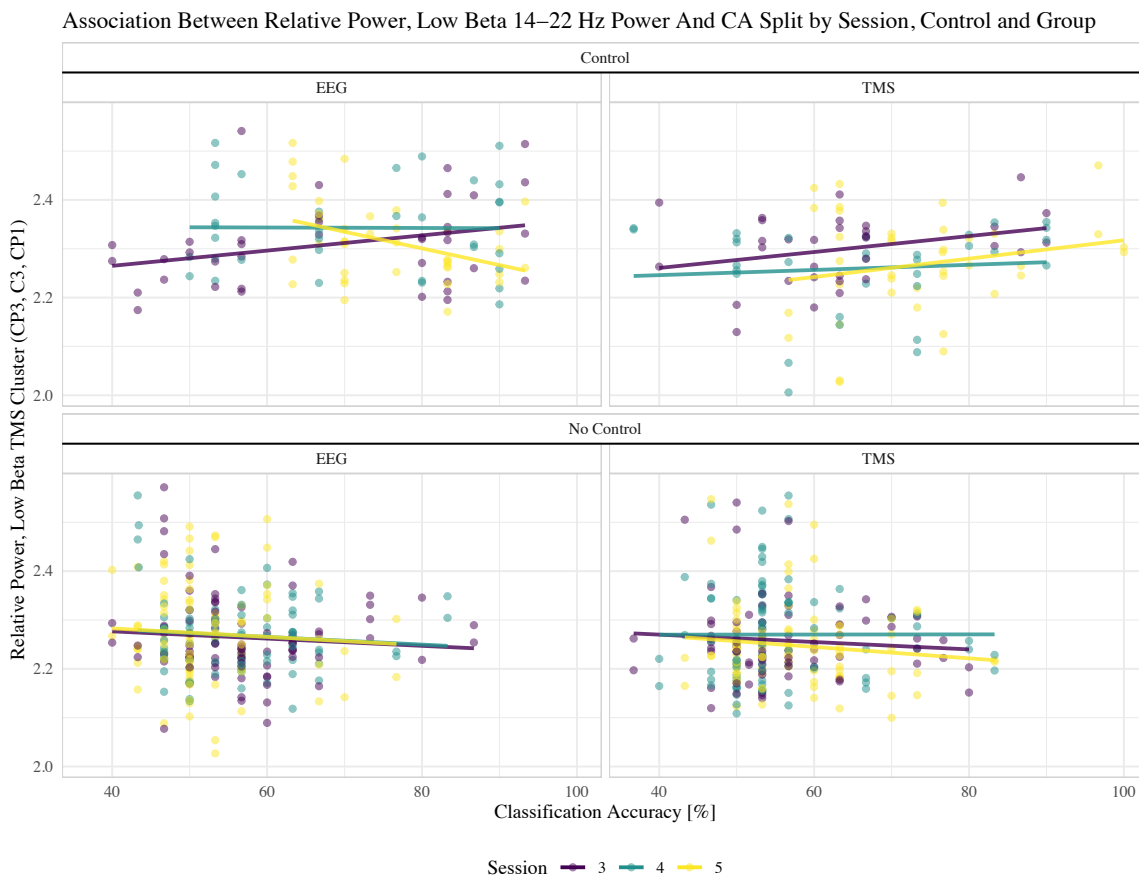


Figure 10.18. Relative Power Low Beta Dotplot

Relative Power, TMS vs EEG Group, Comparison of Third Session vs Last Session

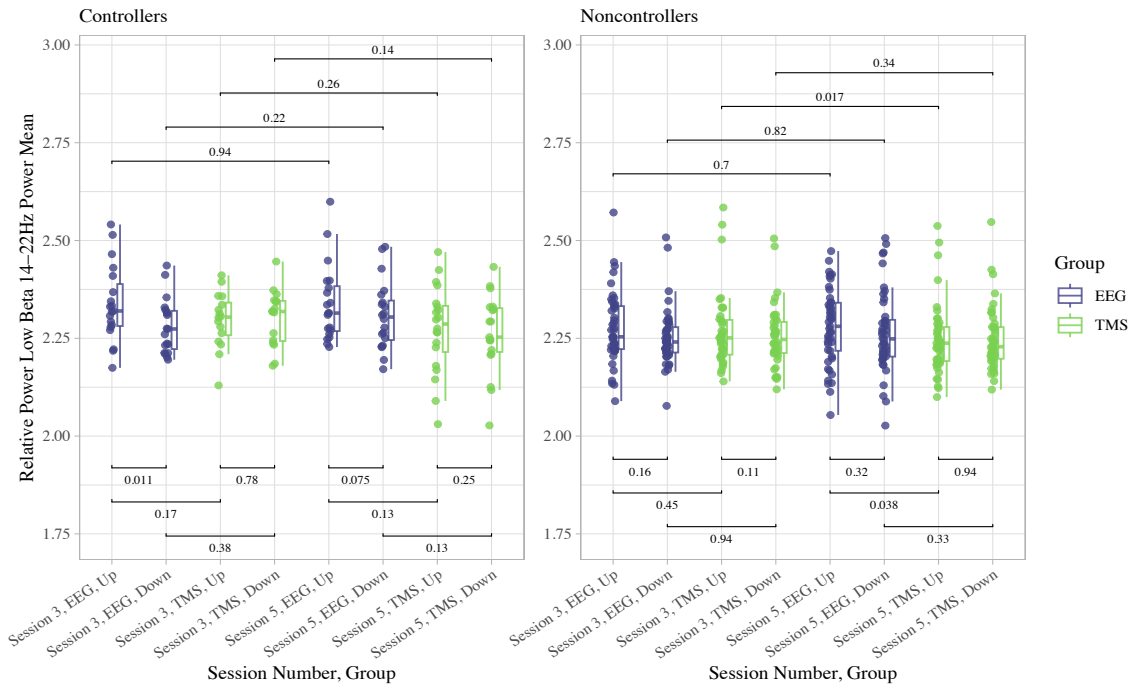


Figure 10.19. Relative Power Low Beta Boxplot

Results for CA, High Delta

Table 10.5

CA Model High Delta

	CA Beta	CA t- Value	CA p- Value	Up Beta	Up t- Value	Up p- Value	Down Beta	Down t- Value	Down p- Value
(Intercept)	70.53	12.68	0.001 >	14.91	6.49	0.001 >	6.5100	2.85	0.0046
Group (TMS)	-9.98	-1.38	0.17	-8.37	-2.81	0.0051	5.5500	1.87	0.0620
Time	-0.69	-2.05	0.04	-0.34	-2.36	0.019	0.1300	0.87	0.3800
High Delta	-0.13	-1.8	0.072	-0.081	-2.65	0.0081	0.0400	1.33	0.1900
Group (TMS):Time	0.58	1.32	0.19	0.59	3.04	0.0025	-0.4400	-2.29	0.0220
Group (TMS):High Delta	0.011	0.13	0.9	0.069	1.83	0.068	-0.0650	-1.76	0.0790
Time:High Delta	0.008	1.75	0.081	0.005	2.48	0.014	-0.0024	-1.20	0.2300
Group (TMS):Time:High Delta	0.00068	0.12	0.91	- 0.0047	-1.86	0.063	0.0051	2.03	0.0430

Association Between Percentage Change, High Delta 1–5 Hz Power And CA Split by Session, Control and Group

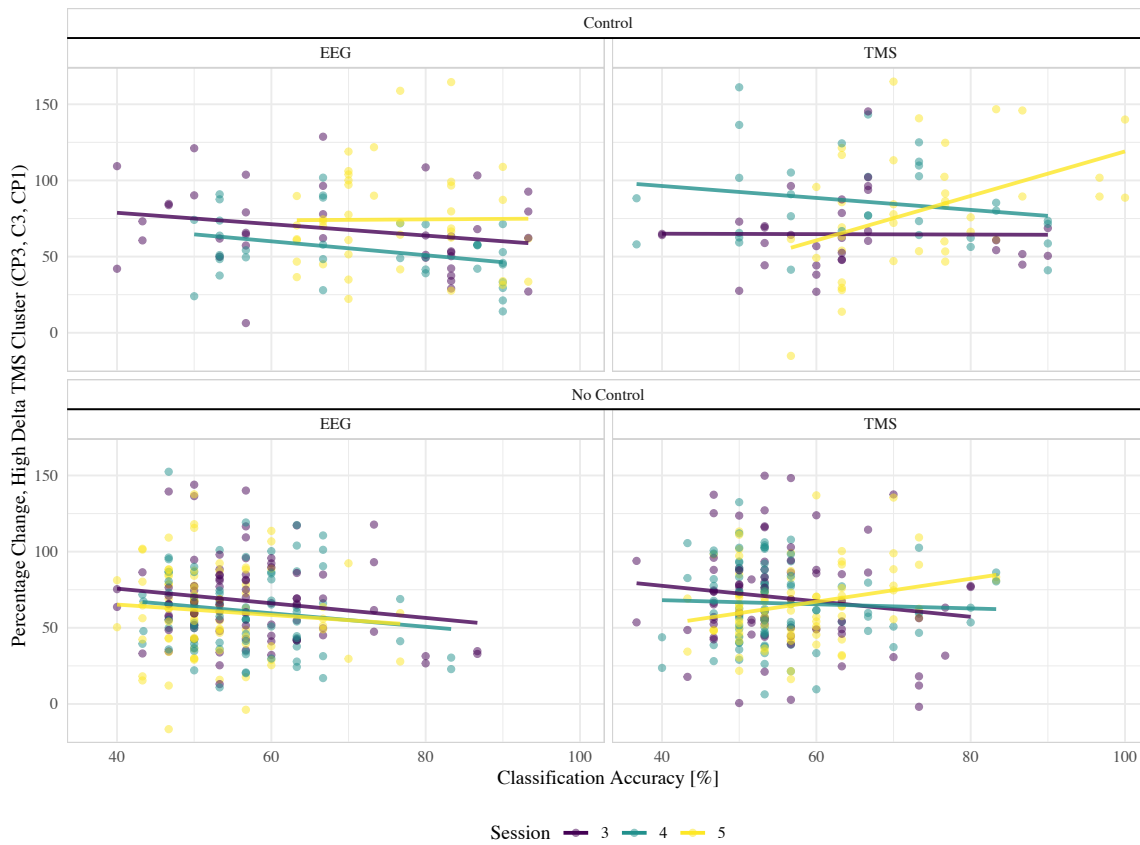


Figure 10.20. Percentage Change High Delta Dotplot

Percentage Change, TMS vs EEG Group, Comparison of Third Session vs Last Session

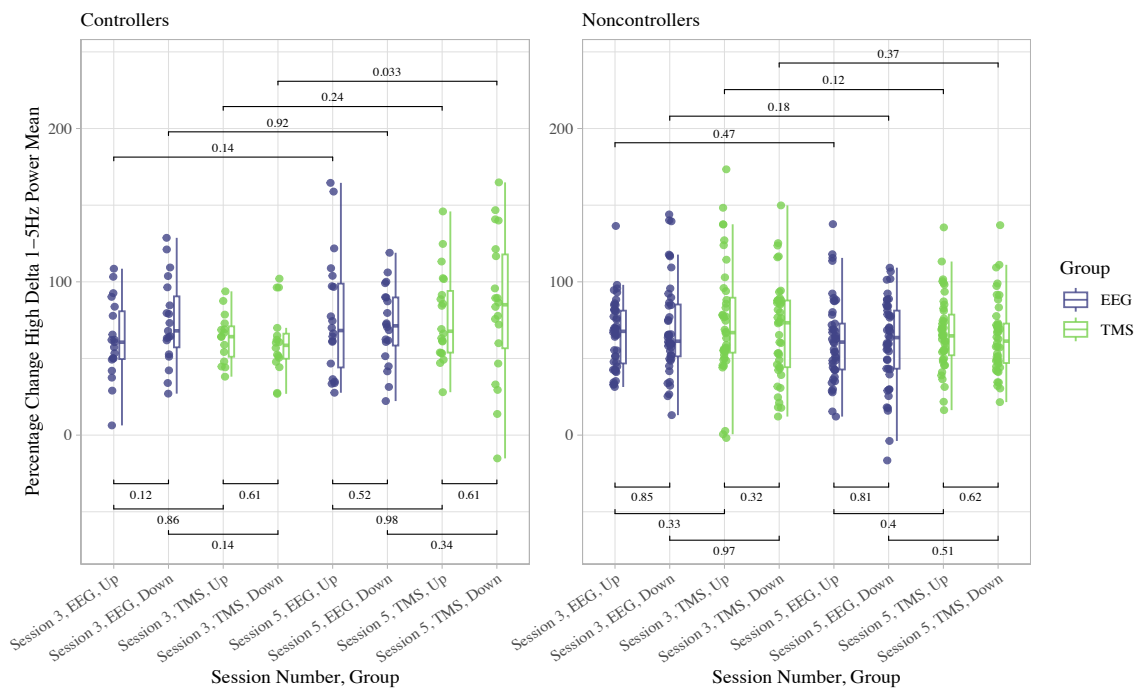


Figure 10.21. Percentage Change High Delta Boxplot

Results for CA, Low Alpha

Table 10.6

CA Model Low Alpha

	CA Beta	CA t-Value	CA p-Value	Up Beta	Up t-Value	Up p-Value	Down Beta	Down t-Value	Down p-Value
(Intercept)	60.95	19.29	0.001 >	10.32	9.34	0.001 >	8	6.99	0.001 >
Group (TMS)	-7.11	-1.56	0.12	-4.7	-2.89	0.0041	2.72	1.62	0.11
Time	-0.093	-0.64	0.52	-0.07	-1.08	0.28	0.041	0.65	0.52
Low Alpha	0.091	1.12	0.26	-0.077	-2.18	0.03	0.1	2.96	0.0032
Group (TMS):Time	0.49	2.28	0.023	0.32	3.32	0.001 >	-0.18	-1.9	0.058
Group (TMS):Low Alpha	-0.14	-1.51	0.13	0.079	1.93	0.053	-0.12	-3.01	0.0027
Time:Low Alpha	-0.0043	-0.76	0.45	0.0052	2.11	0.035	-0.0065	-2.69	0.0073
Group (TMS):Time:Low Alpha	0.0082	1.29	0.2	-0.005	-1.8	0.073	0.0074	2.71	0.0069

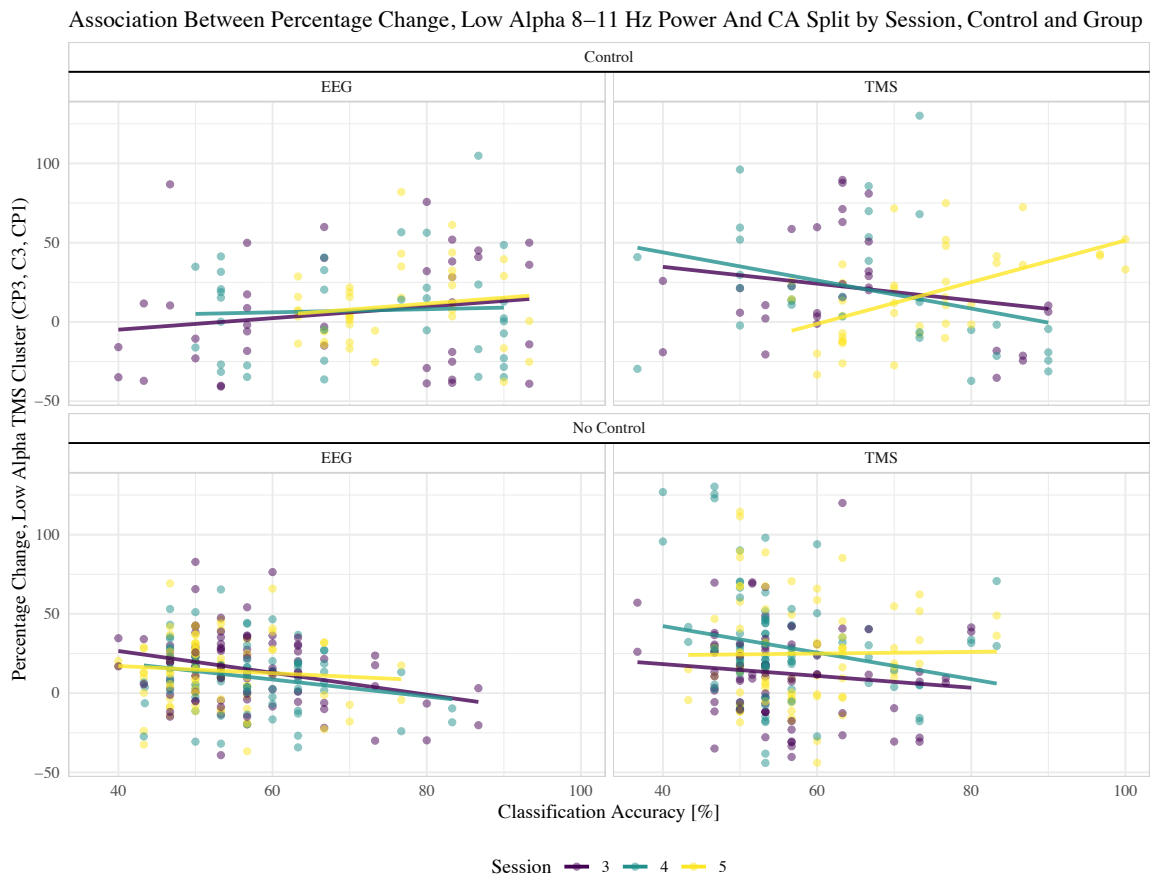


Figure 10.22. Percentage Change Low Alpha Dotplot

Percentage Change, TMS vs EEG Group, Comparison of Third Session vs Last Session

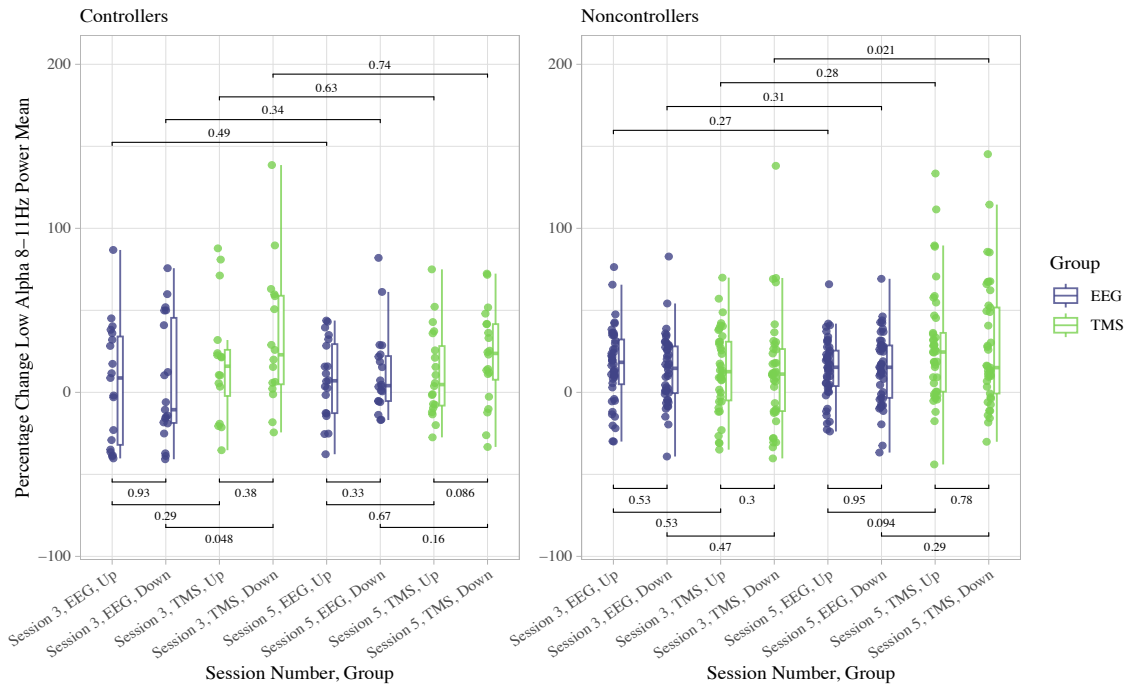


Figure 10.23. Percentage Change Low Alpha Boxplot

Category 3

Results for CA, Theta

Table 10.7

CA Model Theta

	CA Beta	CA t-Value	CA p-Value	Up Beta	Up t-Value	Up p-Value	Down Beta	Down t-Value	Down p-Value
(Intercept)	65	16.73	0.001 >	11.12	7.37	0.001 >	8.49	5.54	0.001 >
Group (TMS)	-2.95	-0.54	0.59	-3.92	-1.83	0.068	3.06	1.41	0.16
Time	-0.33	-1.58	0.11	-0.092	-0.98	0.33	-0.011	-0.12	0.9
Theta	-0.095	-1.17	0.24	-0.051	-1.4	0.16	0.02	0.56	0.58
Group (TMS):Time	0.11	0.37	0.71	0.23	1.74	0.081	-0.21	-1.6	0.11
Group (TMS):Theta	-0.2	-1.87	0.062	0.0023	0.047	0.96	-0.057	-1.19	0.23
Time:Theta	0.0061	1.11	0.27	0.0025	0.97	0.33	-0.00056	-0.22	0.82
Group (TMS):Time:Theta	0.016	2.2	0.028	0.0012	0.35	0.73	0.0036	1.09	0.28

Association Between Percentage Change, Theta 5–8 Hz Power And CA Split by Session, Control and Group

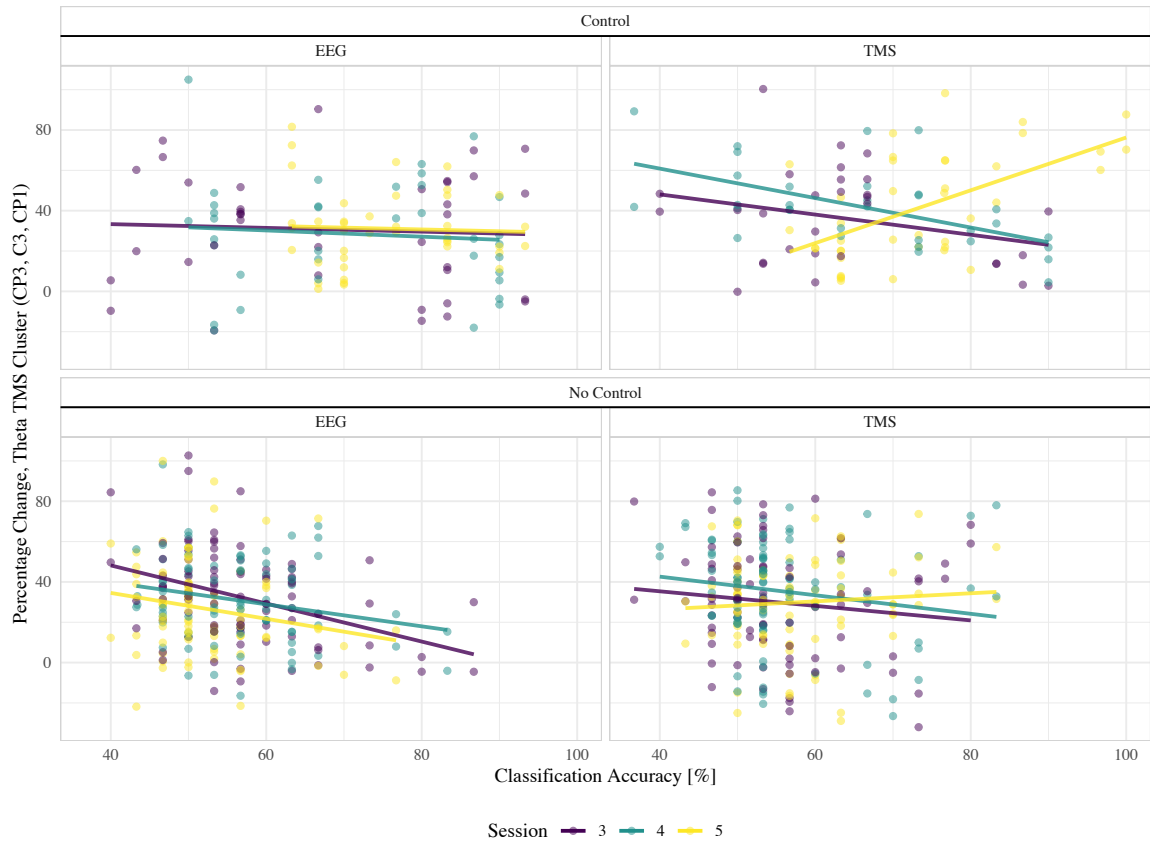


Figure 10.24. Percentage Change Theta Dotplot

Percentage Change, TMS vs EEG Group, Comparison of Third Session vs Last Session

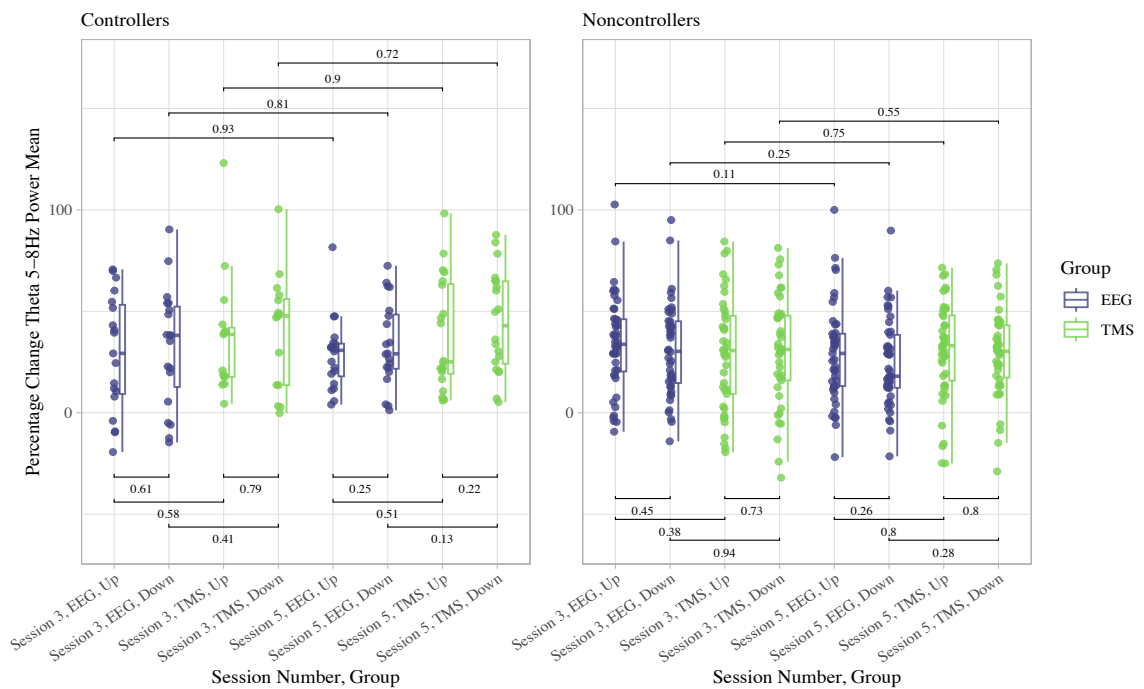


Figure 10.25. Percentage Change Theta Boxplot

Category 4 Results

Results for CA, Low Beta

Table 10.8

CA Model Low Beta

	CA Beta	CA t-Value	CA p-Value	Up Beta	Up t-Value	Up p-Value	Down Beta	Down t-Value	Down p-Value
(Intercept)	62.58	20.55	0.001 >	9.15	8.7	0.001 >	9.59	8.78	0.001 >
Group (TMS)	-9.73	-2.25	0.027	-3.49	-2.32	0.021	0.78	0.5	0.62
Time	-0.17	-1.28	0.2	0.0055	0.091	0.93	-0.056	-0.95	0.34
Low Beta	0.19	1.26	0.21	-0.11	-1.64	0.1	0.16	2.51	0.012
Group (TMS):Time	0.65	3.35	0.001 >	0.25	2.94	0.0034	-0.069	-0.81	0.42
Group (TMS):Low Beta	-0.29	-1.76	0.078	0.078	1.07	0.28	-0.17	-2.32	0.021
Time:Low Beta	-0.013	-1.34	0.18	0.0072	1.75	0.08	-0.011	-2.75	0.0062
Group (TMS):Time:Low Beta	0.019	1.82	0.069	-0.0058	-1.24	0.22	0.012	2.53	0.012



Figure 10.26. Percentage Change Low Beta Dotplot

Percentage Change, TMS vs EEG Group, Comparison of Third Session vs Last Session

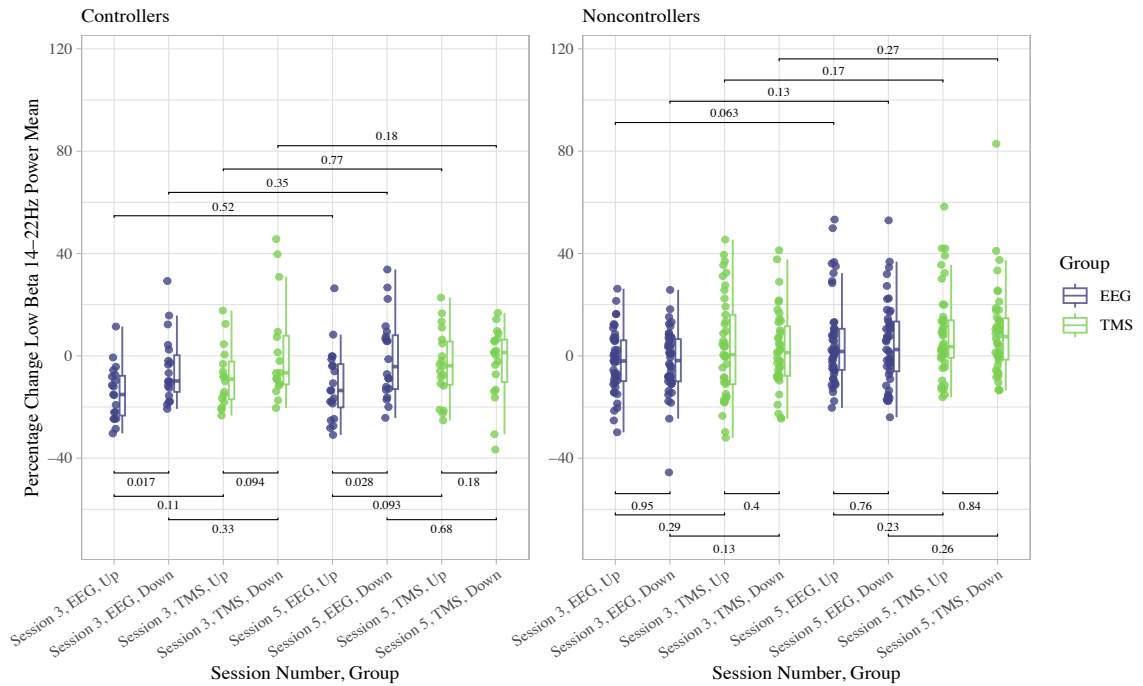


Figure 10.27. Percentage Change Low Beta Boxplot

Category 5 Results

Results for CA, High Beta

Table 10.9

CA Model High Beta

	CA Beta	CA t- Value	CA p- Value	Up Beta	Up t- Value	Up p- Value	Down Beta	Down t- Value	Down p- Value
(Intercept)	61.88	20.42	0.001 >	9.35	8.98	0.001 >	9.19	8.48	0.001 >
Group (TMS)	-8.97	-2.09	0.039	-3.8	-2.54	0.012	1.34	0.86	0.39
Time	-0.14	-1.02	0.31	-0.007	-0.12	0.91	-0.032	-0.56	0.58
High Beta	-0.11	-0.6	0.55	-0.085	-1.05	0.3	0.045	0.57	0.57
Group (TMS):Time	0.62	3.22	0.0014	0.28	3.27	0.0011	-0.11	-1.29	0.2
Group (TMS):High Beta	-0.1	-0.48	0.63	0.086	0.9	0.37	-0.11	-1.21	0.23
Time:High Beta	0.0077	0.65	0.52	0.0064	1.23	0.22	-0.0041	-0.8	0.42
Group (TMS):Time:High Beta	0.0054	0.38	0.7	-0.0084	-1.36	0.18	0.01	1.67	0.096

Association Between Percentage Change, High Beta 22–31 Hz Power And CA Split by Session, Control and Group

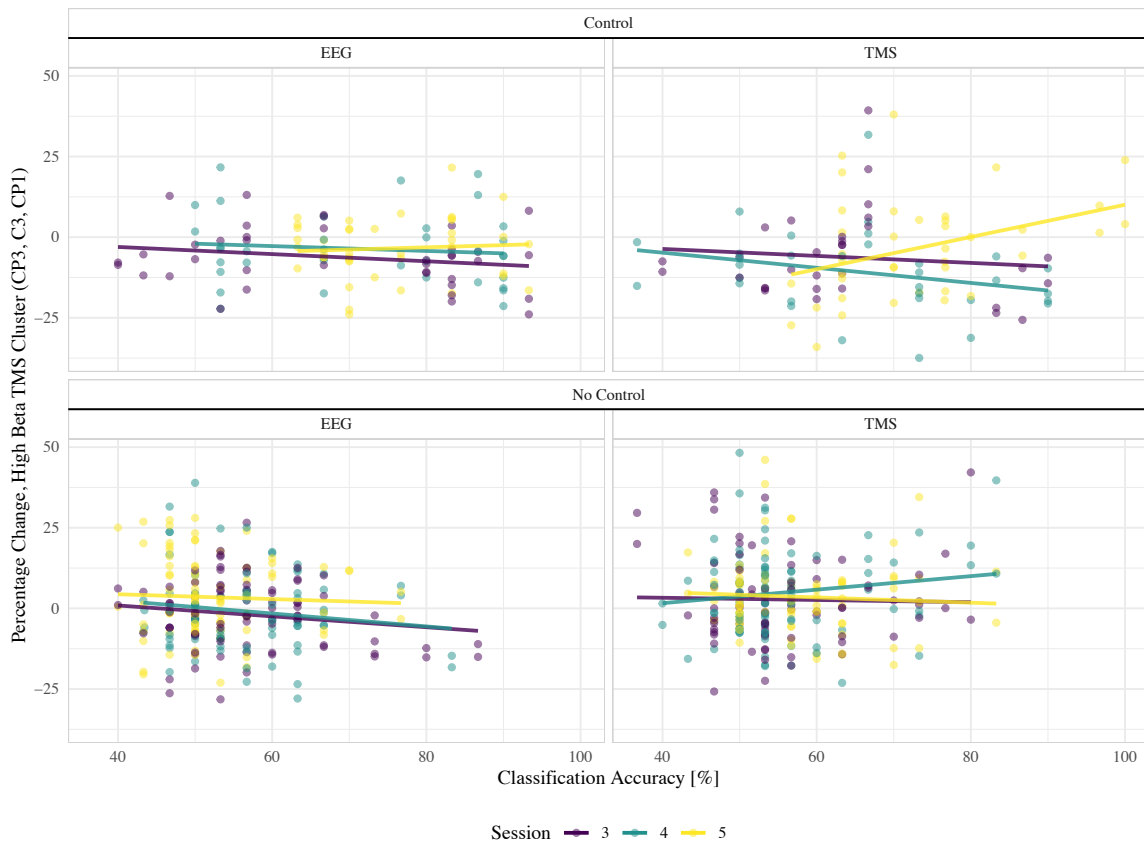


Figure 10.28. Percentage Change High Beta Dotplot

Percentage Change, TMS vs EEG Group, Comparison of Third Session vs Last Session

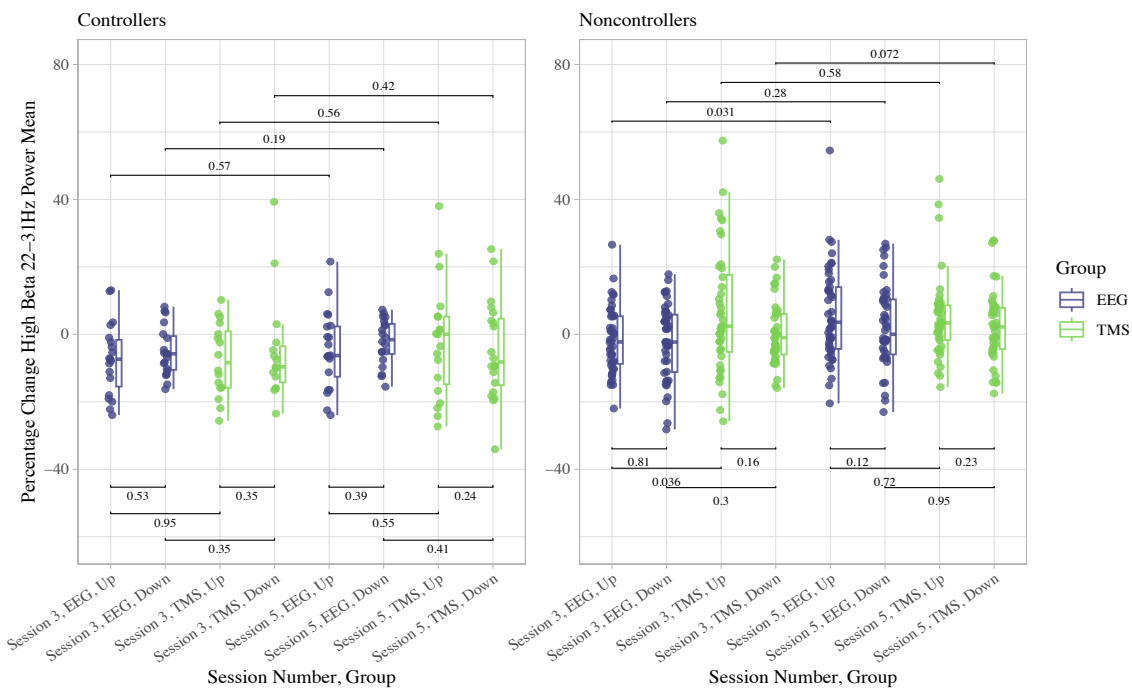


Figure 10.29. Percentage Change High Beta Boxplot

Results for CA, Low Gamma

Table 10.10

CA Model Low Gamma

	CA Beta	CA t-Value	CA p-Value	Up Beta	Up t-Value	Up p-Value	Down Beta	Down t-Value	Down p-Value
(Intercept)	62.67	20.88	0.001 >	9.49	9.21	0.001 >	9.32	8.68	0.001 >
Group (TMS)	-8.77	-2.04	0.044	-3.76	-2.5	0.013	1.35	0.87	0.39
Time	-0.19	-1.46	0.14	-0.012	-0.21	0.83	-0.046	-0.79	0.43
Low Gamma	-0.27	-1.73	0.085	-0.04	-0.58	0.56	-0.046	-0.68	0.49
Group (TMS):Time	0.62	3.2	0.0014	0.27	3.14	0.0017	-0.1	-1.18	0.24
Group (TMS):Low Gamma	0.0096	0.053	0.96	0.018	0.22	0.83	-0.012	-0.16	0.88
Time:Low Gamma	0.019	1.92	0.056	0.0022	0.5	0.62	0.0037	0.86	0.39
Group (TMS):Time:Low Gamma	-0.0037	-0.32	0.75	-0.0015	-0.29	0.78	0.00041	0.081	0.94

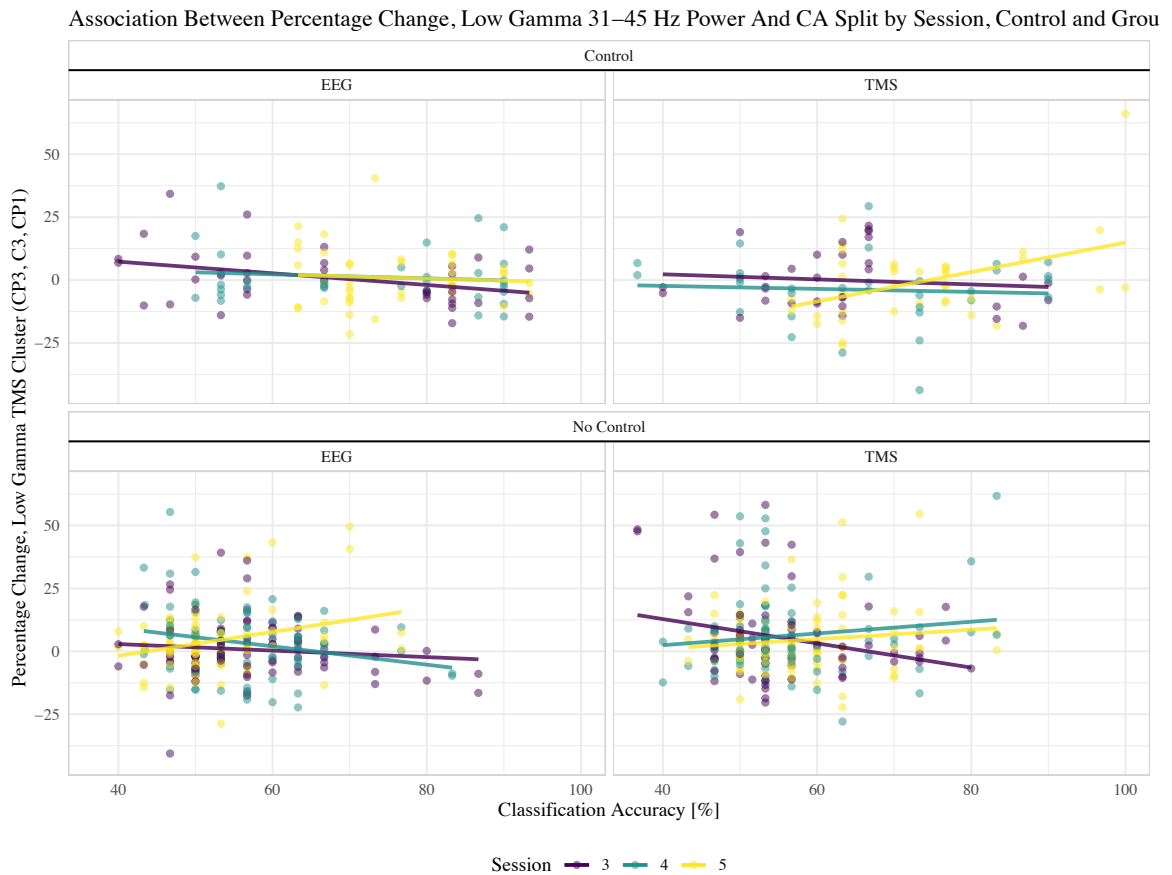


Figure 10.30. Percentage Change Low Gamma Dotplot

Percentage Change, TMS vs EEG Group, Comparison of Third Session vs Last Session

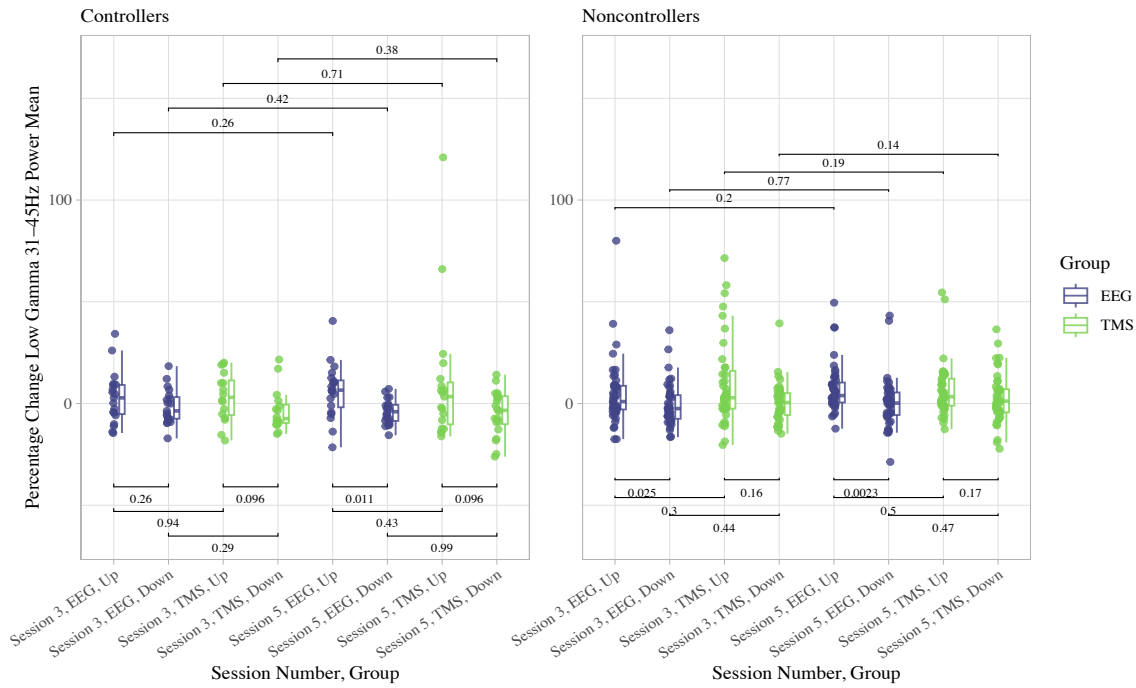


Figure 10.31. Percentage Change Low Gamma Boxplot

Results for CA, Theta

Table 10.11

CA Model Theta

	CA Beta	CA t- Value	CA p- Value	Up Beta	Up t- Value	Up p- Value	Down Beta	Down t- Value	Down p- Value
(Intercept)	41.64	1.03	0.30	-20.02	-1.14	0.260	30.79	1.770	0.077
Group (TMS)	-54.25	-1.09	0.28	-13.10	-0.61	0.550	-1.24	-0.058	0.950
Time	2.26	0.83	0.41	1.63	1.38	0.170	-0.78	-0.670	0.510
Theta	8.85	0.51	0.61	12.63	1.68	0.094	-9.22	-1.240	0.220
Group (TMS):Time	3.01	0.92	0.36	1.72	1.20	0.230	-0.94	-0.660	0.510
Group (TMS):Theta	19.22	0.90	0.37	4.07	0.44	0.660	0.93	0.100	0.920
Time:Theta	-1.04	-0.89	0.37	-0.70	-1.38	0.170	0.32	0.630	0.530
Group (TMS):Time:Theta	-1.02	-0.72	0.47	-0.63	-1.02	0.310	0.37	0.620	0.540



Figure 10.32. Relative Power Theta Dotplot

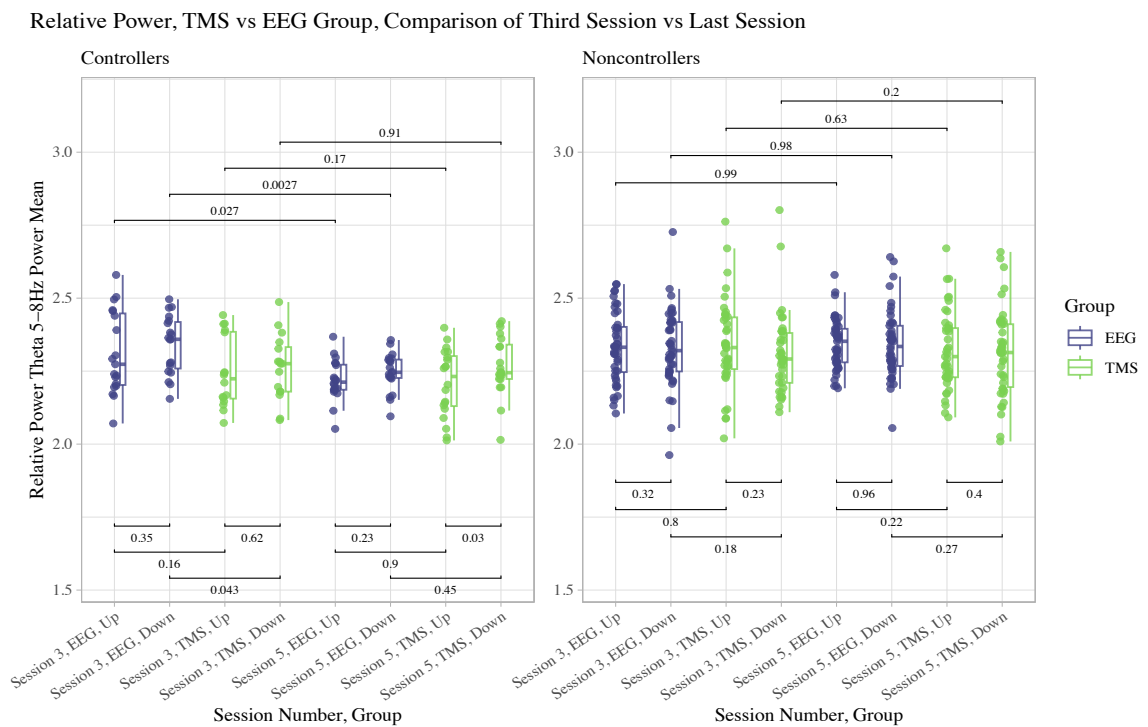


Figure 10.33. Relative Power Theta Boxplot

10.2 Stopping Networks and falling in the elderly, Appendix

10.2.1 Table 1. ROI Comparison

Table 10.12
ROI Voxel Count Comparison

Group	ROI	Voxel Count
ROI	r IFG	13451
	r IFG (sub 1)	5741
	r IFG (sub 2)	4360
	r IFG (sub 3)	3350
ROI	l preSMA	3465
ROI	r preSMA	3173
ROI, Control	r STN	3931
ROI, Control	l STN	3419
Control	l IFG	10495
	l IFG (sub 1)	3083
	l IFG (sub 2)	3241
	l IFG (sub 3)	4171

10.2.2 Assumptions for logistic regression

Assumption of normality for z-standardisation

The ROI connections were normalised using a z-transformation. This was done to increase comparability between regions. For the significant ROI connection of rIFG to rSTN the values of AFD seem normally distributed visually.

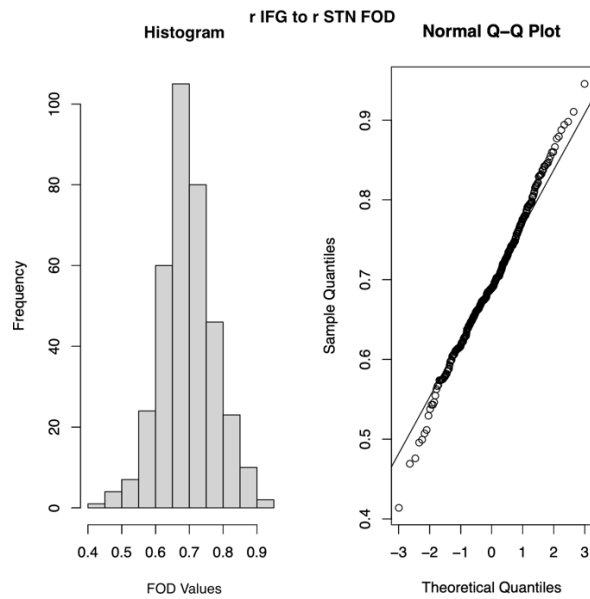


Figure 10.34. rIFG to STN Normal DistributionAFD Values

This is confirmed with the Shapiro-Wilk test for normality, which is not significant $W = 0.99$, $p = 0.17$. However, the Kolmogorov Smirnov test for normality is significant $D = 0.68$, $p < 2.2e-16$. Given that the data visually looks only to have slight departures towards the extremes and overall looks normally distributed. We concluded that the assumption of normal distribution is not rejected.

For the subregion 68 of the rIFG the values also looked normally distributed:

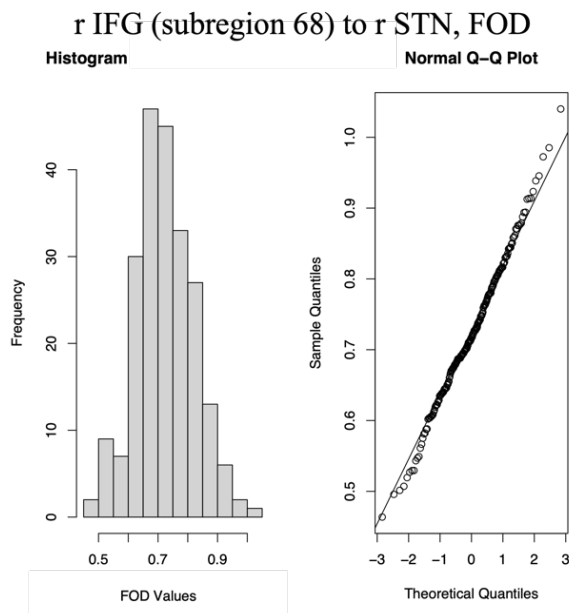


Figure 10.35. rIFG Subregion to STN Normal DistributionAFD Values

This is confirmed with the Shapiro-Wilk test for normality, which is not significant $W = 1$, $p = 0.73$. However, the Kolmogorov Smirnov test for normality is significant $D = 0.69$, $p < 2.2e-16$. Given that the data visually looks only to have slight departures towards

the extremes and overall looks normally distributed. We concluded that the assumption of normal distribution is not rejected.

Assumptions for Logistic Regressions

Overall the assumptions were met for most of the ROI-models. A few ROI-models did not meet criteria. However, the only model we present and interpret is the significant rIFG – rSTN connection and the subregion 68 to rSTN. This ROI-met met all assumptions for logistict regression, as detailed below.

Assumption #1: The Response Variable is Binary

Yes, falls are recorded as 1: fallen, 0: not fallen.

Assumption #2: The Observations are Independent

Should be given. Individual observations are independent (different people).

Assumption #3: There is No Multicollinearity Among Explanatory Variables

This is checked by looking at a metric called General Variance Inflation (GVIF). GVIF below 5 is deemed acceptable, although this is a rule of thumb. Our GVIF are below 5 for all reported models.

Assumption #4: There are No Extreme Outliers

The analysis contains no outliers. Outliers defined as ± 3 standard deviation were excluded.

Assumption #5: Linear relationship between Log of predictor and outcome variable

Given. Plot for the significant result attached:

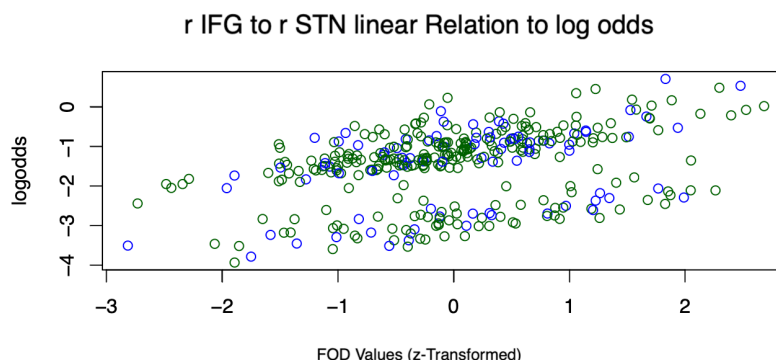


Figure 10.36 Linear relation rIFG

Plot for the subregion 68:

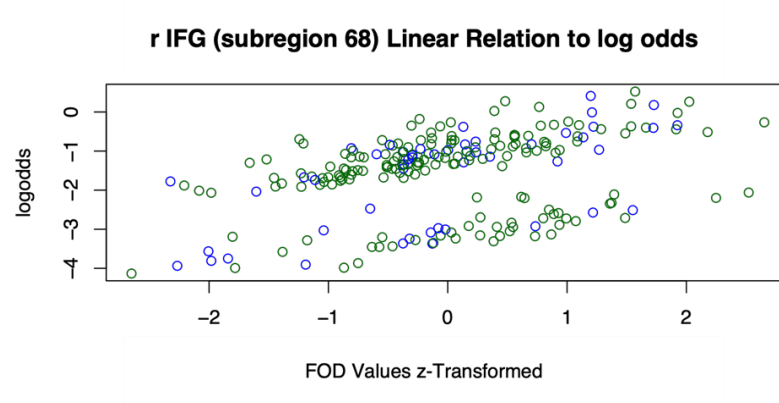


Figure 10.37. Linear Relation rIFG Subregion

Assumption #6: enough n

There should be around 20% of the least likely outcome (fallers, in our case). This is the result for rIFG:

Table 10.13

ROI Cases

ROI	Wave	N Total	N Fallers	N Percentage
rIFG	fallers_w3	360	81	22.5
rIFG Subregion	fallers_w3	221	48	21.72

10.2.3 Results for control area

Table 10.14

Results for Control Area FA and AFD

Associative (Cross-Sectional)

Main ROI, AFD						
Region	Odds	CI Low	CI High	P Value	Chi Square Fit	n
r FFA - r STN	1.56	1.04	2.33	0.03	0.0227	178
lIFG - l STN	0.73	0.53	1	0.05	0.0001	282
lIFG - r STN	0.84	0.59	1.21	0.36	0.0042	197
l FFA - l STN	0.85	0.54	1.35	0.5	0.0050	124
r FFA - l STN	1.34	0.44	4.08	0.61	0.2286	39
l FFA - r STN	0.92	0.42	2	0.83	0.0192	70
SubROI (l IFG), AFD						
l IFG (subregion 41) - l STN	0.67	0.44	1.02	0.06	0.0000	142
l IFG (subregion 60) - l STN	1.28	0.79	2.09	0.31	0.0508	143
l IFG (subregion 64) - l STN	0.81	0.46	1.42	0.46	0.0333	125
l IFG (subregion 60) - r STN	1.28	0.54	3.03	0.57	0.4042	52
l IFG (subregion 64) - r STN	0.88	0.53	1.45	0.62	0.0257	128
l IFG (subregion 41) - r STN	1.16	0.62	2.18	0.64	0.2784	68
Main ROI, FA						
r FFA - r STN	1.5	1.01	2.22	0.05	0.0291	178
lIFG - l STN	0.77	0.56	1.05	0.1	0.0002	282
l FFA - r STN	0.66	0.31	1.39	0.27	0.0129	70
l FFA - l STN	0.83	0.52	1.31	0.42	0.0055	123
lIFG - r STN	0.87	0.6	1.25	0.45	0.0047	197
r FFA - l STN	1.1	0.37	3.23	0.87	0.2433	39
SubROI (l IFG), FA						
l IFG (subregion 41) - l STN	0.57	0.37	0.89	0.01	0.0000	142
l IFG (subregion 60) - l STN	1.55	0.93	2.57	0.09	0.0189	143
l IFG (subregion 64) - r STN	0.81	0.5	1.32	0.4	0.0310	128
l IFG (subregion 60) - r STN	1.27	0.53	3.05	0.59	0.4074	52
l IFG (subregion 41) - r STN	1.13	0.59	2.14	0.71	0.2840	68
l IFG (subregion 64) - l STN	1.09	0.63	1.89	0.75	0.0385	124
Predictive (Wave 3 data predicting wave 4 & 5)						
Main ROI, AFD						

Region	Odds	CI Low	CI High	P Value	Chi Square Fit	n
l IFG - l STN	1.38	1	1.91	0.05	0.0015	223
r FFA - l STN	0.34	0.09	1.3	0.12	0.1150	29
r FFA - r STN	1.37	0.89	2.09	0.15	0.0081	138
l FFA - r STN	1.42	0.73	2.78	0.3	0.2500	55
l FFA - l STN	0.99	0.54	1.84	0.98	0.1232	89
l IFG - r STN	1	0.65	1.53	1	0.1557	147
SubROI (l IFG), AFD						
l IFG (subregion 41) - r STN	2.87	0.91	9.1	0.07	0.0141	49
l IFG (subregion 64) - l STN	1.53	0.92	2.57	0.1	0.0074	102
l IFG (subregion 60) - l STN	1.44	0.91	2.28	0.12	0.0551	113
l IFG (subregion 64) - r STN	1.48	0.86	2.54	0.15	0.1980	100
l IFG (subregion 60) - r STN	0.66	0.25	1.73	0.4	0.5243	36
l IFG (subregion 41) - l STN	1.1	0.66	1.83	0.7	0.0250	104
Main ROI, FA						
r FFA - l STN	0.13	0.02	1.04	0.05	0.0330	29
l IFG - l STN	1.33	0.97	1.82	0.08	0.0020	223
l FFA - l STN	0.76	0.41	1.43	0.4	0.1128	88
r FFA - r STN	1.17	0.77	1.78	0.47	0.0142	138
l FFA - r STN	1.08	0.55	2.09	0.83	0.3269	55
l IFG - r STN	0.98	0.66	1.46	0.94	0.1190	149
Sub ROI, FA						
l IFG (subregion 64) - l STN	1.64	0.95	2.83	0.08	0.0066	101
l IFG (subregion 64) - r STN	1.44	0.81	2.56	0.22	0.2408	99
l IFG (subregion 60) - r STN	0.53	0.19	1.47	0.22	0.4341	36
l IFG (subregion 41) - r STN	1.62	0.64	4.08	0.31	0.0368	49
l IFG (subregion 41) - l STN	1.16	0.69	1.95	0.57	0.0235	104
l IFG (subregion 60) - l STN	0.95	0.63	1.42	0.8	0.1066	114

10.2.4 Results for apparent fibre density

Table 10.15

Results for ROI, AFD

Associative (Cross-Sectional)						
Main ROI, AFD						
Region	Odds	CI Low	CI High	P Value	Chi Square Fit	n
r IFG - r STN	1.49	1.13	1.98	0.005	0.00003	360
l preSMA - r STN	1.38	0.93	2.05	0.113	0.00046	200
r IFG - l STN	1.28	0.91	1.8	0.161	0.00172	237
r preSMA - r STN	0.84	0.61	1.16	0.288	0.02334	257
r preSMA - l STN	0.91	0.65	1.26	0.557	0.00976	228
l preSMA - l STN	0.97	0.73	1.29	0.855	0.00096	343
SubROI (l IFG), AFD						
r IFG (subregion 79) - l STN	1.68	1.07	2.65	0.024	0.011	130
r IFG (subregion 68) - r STN	1.52	1.05	2.2	0.026	0.002	221
r IFG (subregion 79) - r STN	1.26	0.91	1.75	0.17	0.001	249
r IFG (subregion 68) - l STN	0.88	0.53	1.46	0.625	0.060	139
r IFG (subregion 90) - r STN	0.92	0.63	1.35	0.671	0.049	186
r IFG (subregion 90) - l STN	1.2674E+13	0	> 1.2674E+13	1	0.000	35
Predictive (Wave 3 data predicting wave 4 & 5)						
Main ROI, AFD						
l preSMA - l STN	1.42	1.05	1.92	0.023	0.023	265
r preSMA - r STN	0.74	0.52	1.06	0.097	0.017	200
r preSMA - l STN	0.81	0.54	1.2	0.29	0.026	171
r IFG - r STN	1.04	0.78	1.39	0.776	0.125	276
r IFG - l STN	0.97	0.66	1.42	0.861	0.330	180
l preSMA - r STN	0.98	0.69	1.4	0.926	0.387	156
SubROI (l IFG), AFD						
r IFG (subregion 68) - l STN	0.72	0.43	1.18	0.19	0.589	114
r IFG (subregion 90) - r STN	1.15	0.78	1.71	0.473	0.094	146
r IFG (subregion 68) - r STN	0.91	0.64	1.29	0.601	0.065	173
r IFG (subregion 79) - r STN	1.04	0.74	1.47	0.801	0.128	192

r IFG (subregion 79) - 1 STN	1.05	0.63	1.73	0.861	0.121	98
r IFG (subregion 90) - 1 STN	0	0	NA	1	0.007	27

10.2.5 *Results for fractional anisotropy*

Table 10.16

Results for ROI, FA

Associative (Cross-Sectional)

Main ROI, FA						
Region	Odds	CI Low	CI High	P Value	Chi Square Fit	n
l preSMA - r STN	1.42	0.96	2.1	0.077	0.002	201
r IFG - l STN	1.31	0.93	1.83	0.119	0.001	237
r preSMA - l STN	0.79	0.57	1.11	0.173	0.007	229
r preSMA - r STN	0.82	0.6	1.11	0.193	0.029	260
r IFG - r STN	1.2	0.9	1.59	0.22	0.000	362
l preSMA - l STN	1.01	0.77	1.33	0.924	0.001	347
SubROI (l IFG), FA						
r IFG (subregion 79) - l STN	1.78	1.13	2.79	0.012	0.007	130
r IFG (subregion 68) - r STN	1.44	0.98	2.11	0.06	0.008	222
r IFG (subregion 79) - r STN	1.18	0.83	1.68	0.352	0.002	250
r IFG (subregion 90) - r STN	0.88	0.6	1.3	0.53	0.046	186
r IFG (subregion 68) - l STN	0.94	0.57	1.54	0.806	0.063	140
r IFG (subregion 90) - l STN	2.525E+10	0	> 2.525E+10	1	0.000	35
Predictive (Wave 3 data predicting wave 4 & 5)						
Main ROI, FA						
l preSMA - l STN	1.52	1.14	2.03	0.005	0.017	259
r preSMA - r STN	0.83	0.58	1.17	0.279	0.050	192
r preSMA - l STN	0.88	0.59	1.31	0.519	0.057	167
l preSMA - r STN	1.1	0.77	1.57	0.606	0.420	152
r IFG - r STN	1.06	0.79	1.42	0.703	0.166	271
r IFG - l STN	1.05	0.72	1.54	0.783	0.347	178
Sub ROI, FA						
r IFG (subregion 68) - r STN	0.81	0.57	1.16	0.246	0.120	168
r IFG (subregion 68) - l STN	0.77	0.47	1.26	0.3	0.542	109
r IFG (subregion 79) - r STN	0.86	0.59	1.26	0.432	0.149	187
r IFG (subregion 90) - l STN	0.48	0.04	6.25	0.577	0.704	25

r IFG (subregion 79) - l STN	0.96	0.58	1.6	0.889	0.090	96
r IFG (subregion 90) - r STN	1.02	0.69	1.52	0.918	0.162	140

10.2.6 Results for Apparent Fibre Density with Physical Activity interaction

Table 10.17

Results for AFD with Physical Activity

Associative (Cross Sectional)						
Main, AFD						
Region	Odds	CI Low	CI High	p-Value	Chi Square Fit	n
rIFG - r STN	2.31	1.42	3.78	0.00082	0.000061	360
l preSMA - r STN	1.31	0.74	2.3	0.35	0.0098	200
l preSMA - l STN	1.23	0.76	2	0.41	0.0037	343
r preSMA - l STN	0.8	0.43	1.49	0.48	0.041	228
r preSMA - r STN	0.83	0.47	1.47	0.53	0.1	257
rIFG - l STN	1.06	0.61	1.86	0.83	0.0083	237
Sub, AFD						
rIFG (subregion 68) - r STN	3.59	1.7	7.56	0.00079	0.000027	221
rIFG (subregion 79) - r STN	2.08	1.06	4.05	0.032	0.007	249
rIFG (subregion 79) - l STN	1.79	0.84	3.8	0.13	0.16	130
rIFG (subregion 90) - r STN	0.58	0.28	1.19	0.14	0.21	186
rIFG (subregion 68) - l STN	1.22	0.45	3.31	0.7	0.0016	139
rIFG (subregion 90) - l STN	8.93E+21	0	-	1	0.0014	35
Predictive (Wave 3 data predicting wave 4 & 5)						
Main, AFD						
r preSMA - r STN	0.43	0.22	0.84	0.014	0.039	201
l preSMA - l STN	1.55	0.93	2.58	0.09	0.14	268
rIFG - r STN	1.16	0.74	1.84	0.52	0.28	282
rIFG - l STN	0.92	0.5	1.7	0.79	0.34	186
l preSMA - r STN	1.04	0.64	1.69	0.88	0.24	158
r preSMA - l STN	1.03	0.5	2.12	0.94	0.12	173
Sub, AFD						
rIFG (subregion 79) - r STN	1.26	0.67	2.36	0.47	0.31	193
rIFG (subregion 79) - l STN	0.73	0.3	1.75	0.48	0.12	99
rIFG (subregion 68) - l STN	0.93	0.42	2.09	0.86	0.47	114

rIFG (subregion 90) - r STN	0.98	0.48	2.01	0.96	0.29	146
rIFG (subregion 68) - r STN	1.01	0.57	1.76	0.98	0.13	173
rIFG (subregion 90) - l STN	5.72E+23	0	-	1	0.056	27

10.2.7 Results for Fractional Anisotropy with Physical Activity interaction

Table 10.18

Results for FA with Physical Activity

Associative (Cross Sectional)						
FA Main						
Region	Odds	CI Low	CI High	p-Value	Chi Square Fit	n
r preSMA - r STN	0.48	0.25	0.93	0.03	0.056	260
r IFG - r STN	1.36	0.86	2.15	0.18	0.0026	362
l preSMA - r STN	1.5	0.8	2.79	0.21	0.034	201
r preSMA - l STN	0.73	0.4	1.34	0.31	0.031	229
l preSMA - l STN	1.11	0.69	1.78	0.66	0.0042	347
r IFG - l STN	0.93	0.53	1.65	0.81	0.0061	237
FA Sub						
Region	Odds	CI Low	CI High	p-Value	Chi Square Fit	n
r IFG (subregion 68) - r STN	3.68	1.59	8.52	0.0023	0.000081	222
r IFG (subregion 90) - r STN	0.42	0.19	0.92	0.03	0.079	186
r IFG (subregion 79) - l STN	1.43	0.68	3.01	0.34	0.11	130
r IFG (subregion 79) - r STN	0.83	0.47	1.49	0.54	0.03	250
r IFG (subregion 68) - l STN	1.1	0.46	2.66	0.83	0.002	140
r IFG (subregion 90) - l STN	4.47E-14	0	-	1	0.0014	35
Predictive (Wave 3 data predicting wave 4 & 5)						
FA, Main						
Region	Odds	CI Low	CI High	p-Value	Chi Square Fit	n
l preSMA - l STN	2.01	1.17	3.45	0.011	0.039	268
r preSMA - r STN	0.46	0.21	0.99	0.048	0.083	201
r IFG - r STN	1.35	0.83	2.22	0.23	0.22	282
l preSMA - r STN	1.17	0.68	2.02	0.57	0.095	158
r preSMA - l STN	1.17	0.58	2.37	0.65	0.17	173
r IFG - l STN	1.08	0.58	2.02	0.8	0.34	186
FA, Sub						
Region	Odds	CI Low	CI High	p-Value	Chi Square Fit	n

r IFG (subregion 90) - r STN	0.63	0.29	1.36	0.24	0.23	146
r IFG (subregion 79) - l STN	0.71	0.29	1.77	0.46	0.083	99
r IFG (subregion 79) - r STN	1.22	0.65	2.3	0.54	0.21	193
r IFG (subregion 68) - r STN	0.83	0.46	1.51	0.55	0.13	173
r IFG (subregion 68) - l STN	1.25	0.53	2.91	0.61	0.57	114
r IFG (subregion 90) - l STN	2.11E-124	0	-	1	0.056	27

10.3 Ethics Approval



Coláiste na Tríonóide, Baile Átha Cliath
Trinity College Dublin

Ollscoil Átha Cliath | The University of Dublin

F.A.O. Colin Simon

Approval ID: SPREC012020-14

School of Psychology Research Ethics Committee

29th May 2020

Dear Colin,

The School of Psychology Research Ethics Committee has reviewed your application entitled “A two-stage process for improving Brain-Computer Interface outcomes”, and I am pleased to inform you that it was approved.

Please note that you will be required to submit a completed **Project Annual Report Form** on each anniversary of this approval. Copies of the form are available for download from the Ethics section of the School website.

Please note that you must be familiar with and adhere to the attached ‘Safety Protocol for Adults’.

Adverse events associated with the conduct of this research must be reported immediately to the Chair of the Ethics Committee.

Yours sincerely,

A handwritten signature in black ink, appearing to read 'Richard Carson'.

Richard Carson

Chair,

School of Psychology Research Ethics Committee

10.4 Code

The following MATLAB and R Scripts are what was used to investigate the effects reported in the empirical chapters.

Much of the TMS-NF code is not publicly available and is therefore not listed here.

The most important script is the “Useful Functions” as it contained much of the code needed to investigate the data.

10.4.1 *Useful Functions (R)*

This script contains much of the code used in other R scripts to preprocess, investigate, and plot data.

```
# This function will add a smaller df to a larger df
# it will check for columns unique to the smaller df and add empty columns with
# that name to the larger df. Then it will check a Unique ID in the smaller df
# and if it finds a match in the larger df, the values of the unique columns to
# the larger df. if any rows are not matched, it reports so.
merge_unique_cols <- function(larger_df, smaller_df, uid_col){

  # check if uid_col exists in both data frames
  if(!(uid_col %in% names(larger_df)) || !(uid_col %in% names(smaller_df))){
    stop("unique identifier column not found in one or both data frames")
  }

  # check column names and add unique columns of smaller dataframe to the larger dataframe
  common_cols <- intersect(names(larger_df), names(smaller_df))
  new_cols <- setdiff(names(smaller_df), common_cols)

  # add new columns to larger dataframe if any
  if(length(new_cols) > 0){
    for(col in new_cols){
      larger_df[[col]] <- NA
    }
  }

  # loop over rows of smaller dataframe and append to larger dataframe if uid matches
  collect_warn = character()
  for(i in 1:nrow(smaller_df)){
    uid_match <- as.character(larger_df[[uid_col]]) ==
as.character(smaller_df[[uid_col]][i])
    if(any(uid_match)){
      for(col in new_cols){
        larger_df[uid_match, col] <- smaller_df[i, col]
      }
    } else {
      warn_message = paste("Found no match for UID: ", df.ca$UID[i])
      warn(message = warn_message)
      collect_warn[length(collect_warn) + 1] = as.character(df.ca$UID[i])
    }
  }

  # return warnings as well as df
  if (length(collect_warn > 0)){
    result = list()
    result$warnings = collect_warn
  }
```

```

    result$dataframe = larger_df
  }else{
    result = larger_df
  }

  # return final dataframe
  return(result)
}

# trim to exclude outliers (sd)
ID.outliers = function(data.vector, multiplier = 3){
  # outliers are whatever is multiplier times bigger than mean
  data.mean = mean(data.vector, na.rm = TRUE)
  data.sd = sd(data.vector, na.rm = TRUE)
  upper.thresh = data.mean + data.sd*multiplier
  lower.thresh = data.mean - data.sd*multiplier
  result = data.vector > upper.thresh | data.vector < lower.thresh
  return(result)
}

# Chi Square test and report
ChiSquareTest <- function(data.matrix, data.var, group.var, test.name)
{
  result = list()
  result$name = test.name
  lev1 = levels(as.factor(data.matrix[, group.var]))[1]
  lev2 = levels(as.factor(data.matrix[, group.var]))[2]
  t.x = table(data.matrix[data.matrix[, group.var] == lev1, data.var])
  t.y = table(data.matrix[data.matrix[, group.var] == lev2, data.var])
  t.matrix = matrix(c(t.x, t.y), nrow = length(t.x))
  rownames(t.matrix) = rownames(t.x)
  colnames(t.matrix) = c(lev1, lev2)
  result$matrix = t.matrix
  result$test = chisq.test(t.matrix)
  return(result)
}

IndependantTest <- function(x, y, testName, directionality = "two.sided"){
  # Flag variable for non-parametric procedure
  flag_nonpara = FALSE

  # omit na values
  x = na.omit(x)
  y = na.omit(y)

  # create results list
  result = list()

  # test for normal distributions
  x_norm = shapiro.test(x)
  y_norm = shapiro.test(y)
  # if not normal raise flag
  if ((x_norm$p.value < 0.05) | (y_norm$p.value < 0.05)){
    flag_nonpara = TRUE
  }

  # check homogeneity of variance
  res.ftest = var.test(x,y)
  # if not same raise flag
  if (res.ftest$p.value < 0.05){
    flag_nonpara = TRUE
  }

  # do nonparametric test if flag raised
  if (flag_nonpara){
    result.parameters = list("Distributions non-parametric, Wilcox Test used.", x_norm,
y_norm, res.ftest)

```

```

    result.test = wilcox.test(x, y, alternative = directionality)
  }else{ # else do the normal test
    result.parameters = list("Distributions parametric, t-Test used.", x_norm, y_norm,
res.ftest)
    result.test = t.test(x, y, var.equal = TRUE, alternative = directionality)
  }

  # collect stats for report
  result.mean.x = signif(mean(x, na.rm = TRUE), 4)
  result.sd.x = signif(sd(x, na.rm = TRUE), 4)
  result.med.x = signif(median(x, na.rm = TRUE), 4)
  result.mean.y = signif(mean(y, na.rm = TRUE), 4)
  result.sd.y = signif(sd(y, na.rm = TRUE), 4)
  result.med.y = signif(median(y, na.rm = TRUE), 4)
  result.phrase.x = paste("Mean x = ", result.mean.x, ", SD x = ", result.sd.x, ", Median x
= ", result.med.x, sep = "")
  result.phrase.y = paste("Mean y = ", result.mean.y, ", SD y = ", result.sd.y, ", Median y
= ", result.med.y, sep = "")
  result.phrase = list(result.phrase.x, result.phrase.y)

  # give back list of results
  result$testName = testName
  result$parameters = result.parameters
  result$phrase = result.phrase
  result$test = result.test

  return(result)
}

PairedTest <- function(x, y, testName, directionality = "two.sided"){
  result = list()

  result$name = testName
  # check if differences are normally distributed:
  if (length(na.omit(c(x-y))) > 5){
    norm = shapiro.test(c(x-y))
    result$ShapiroWilk_Norm = norm
  }else{
    norm = list()
    norm$p.value = 0
    norm$expl = "Sample size is too small for normality. Assume non-normal distribution."
    result$ShapiroWilk_Norm = norm
  }

  # get mean, med, sd of data
  mean_x = signif(mean(x, na.rm = TRUE), 4)
  med_x = signif(median(x, na.rm = TRUE), 4)
  sd_x = signif(sd(x, na.rm = TRUE), 4)
  n_x = length(na.omit(x))
  result$x_summ = paste("N = ", n_x, " Mean = ", as.character(mean_x), ", SD = ",
as.character(sd_x),
                        ", Median = ", med_x, sep = "")

  mean_y = signif(mean(y, na.rm = TRUE), 4)
  med_y = signif(median(y, na.rm = TRUE), 4)
  sd_y = signif(sd(y, na.rm = TRUE), 4)
  n_y = length(na.omit(y))
  result$y_summ = paste("N = ", n_y, " Mean = ", as.character(mean_y), ", SD = ",
as.character(sd_y),
                        ", Median = ", med_y, sep = "")

  # check if normal
  if (norm$p.value < 0.05){
    # not normal, do paired wilcoxon test
    result$test = wilcox.test(x, y, paired = TRUE, alternative = directionality)
  }else{
    # is normal, do paired t test

```

```

    result$test = t.test(x, y, paired = TRUE, alternative = directionality)
  }
  return(result)
}

MakePaired <- function(data1, data2, uid_column) {
  # Merge the two data frames on the UID column
  result <- merge(data1, data2, by = uid_column, all = TRUE)

  # Identify rows that do not have a result in both data frames
  excluded <- result[!complete.cases(result), ]

  # Exclude any row of the result data frame that does not have a result in both data
frames
  result <- result[complete.cases(result), ]

  # Return a list containing both data frames
  return(list(result = result, excluded = excluded))
}

# Check summary stats of a variable (typically newly created)
check_summary <- function(variable){
  result = list()
  print(substitute(variable))
  print("Mean:")
  print(mean(variable, na.rm = TRUE))
  print("Standard Deviation:")
  print(sd(variable, na.rm = TRUE))
  print("Range:")
  print(range(variable, na.rm = TRUE))
  print("Table:")
  table = capture.output(table(variable))
  print(table[2:3])
  print("Quantiles:")
  print(quantile(variable))
  plot(variable)
}

# Regression Function -----
# Omit Outliers
omit.outliers = function(data.vec, Omit = TRUE, outliers = NULL, outlier_sd_range = 3){
  # does a vector with outliers ID exist?
  if (is.null(outliers)){
    outliers = ID.outliers(data.vector = data.vec, outlier_sd_range)
  }
  # Omit means just pretend they never exist
  if (Omit){
    result = data.vec[!outliers]
    # if not omit then pretend they are missing values
  } else {
    result = data.vec
    result[outliers] = NA
  }
  return(result)
}

# z transform stuff
z.transform = function(data.vec){
  data.vec.mean = mean(data.vec, na.rm = TRUE)
  data.vec.sd = sd(data.vec, na.rm = TRUE)
  result = (data.vec - data.vec.mean)/data.vec.sd
  return(result)
}

# Checking the things I want to check in one function
Model_me_this <- function(model.data, model.formula, save_path, plot.model = TRUE){

```

```

Results = list()
Results$Formula = formula(model.formula)
Results$save_dir = save_path

# Check if path does not exist
if (!exists(dirname(Results$save_dir))) {
  # create path then
  dir.create(dirname(Results$save_dir))
}
Results$LM = Basic_LM(lm.data = model.data,
                      lm.formula = Results$Formula,
                      save_path = Results$save_dir,
                      Save_LM = TRUE)

if(plot.model){
  Results$Plot = classic.CA.Plot(model.data,
                                lm.model = Results$LM$Model,
                                save_path = Results$save_dir)
}
Results$Ass = check_lm_assumptions(Results$LM$Model,
                                   save_path = Results$save_dir)

return(Results)
}

# Create some basic LM plots (for now only lmer)
Basic_LM <- function(lm.data, lm.formula, save_path, variables = NA, Save_LM = TRUE){
  Results = list()
  Results$Model.formula = formula_to_string(lm.formula)

  # create Model
  lm.model = lmer(lm.formula, data = lm.data)
  Results$Model = lm.model

  # Model summary
  Results$Summary = summary(lm.model)

  # are there variables of interest?
  # if yes collect them
  Results$Vars = NA
  var_specs = NA
  if (!is.na(variables)){
    var_specs = list()
    for (var in variables) {
      var_specs$pval[[var]] = round(summary(chunk.model)$coefficients[var, "Pr(>|t|)"], 4)
      var_specs$coeff[[var]] = round(summary(chunk.model)$coefficients[var, "Estimate"], 4)
    }
  }
  Results$Vars = var_specs

  # Save the whole effort
  # save LM
  if (Save_LM) {
    Save_lm_summ(lm.model, File_Name = save_path)
  }

  # save results
  saveRDS(Results, file = paste0(save_path, "-Var-Output.RData"))
  write(capture.output(Results), file = paste0(save_path, "-Var-Output.txt"))

  return(Results)
}

# Classic CA plot:
classic.CA.Plot <- function(model.data, lm.model, save_path){
  require(ggh4x)
  # model.data = df.EEGBCI
  # lm.model = lm.base$Model
  # save_path = "../Processed Data/StabDist/Models/Base"
  # check dependent variable

```

```

y.var = strsplit(formula_to_string(formula(lm.model)), split = " ~")[[1]][1]

# create plot
gg.modelplot = ggplot(model.data, aes(x = Continuous.Block, y = model.data[[y.var]],
colour = Group)) +
  facet_nested_wrap(~Group + PID, nrow = 2) +
  geom_point(alpha = 0.5) +
  theme_minimal() +
  theme(text = element_text(family = "Times"),
        legend.position = "none",
        panel.spacing = unit(0.2, "lines")) +
  geom_line(data = cbind(model.data, pred = predict(lm.model)), aes(y = pred), size = 1)
+
  labs(x = "Training Blocks", y = paste0(y.var),
        title = "Improvements of Individual Participants Over time") +
  scale_colour_viridis_d(alpha = 0.8, direction = 1) +
  scale_fill_viridis_d(alpha = 0.8, direction = 1)

# save plot
ggsave(filename = paste0(save_path, "-Classic-", y.var, "-Plot.pdf"), width = 10, height =
4.902)
plot(gg.modelplot)

# return plot
return(gg.modelplot)
}

# Check assumptions of LM
check_lm_assumptions <- function(model, save_path, model.name = "default.model.name") {
  require(car)
  require(ggpubr)

  Results = list()

  Results$Model.formula = formula_to_string(formula(model))

  # give it a descriptive model name if none given
  if (model.name == "default.model.name"){
    model.name = formula_to_string(formula(model))
    model.fullname = model.name
    model.name = gsub("\\+", "-p-", model.name)
    model.name = gsub("\\*", "-t-", model.name)
    model.name = gsub("\\|", "-or-", model.name)
    if (nchar(model.name) > 40){
      model.name = substr(model.name, 1, 30)
    }
  }
  rand_int = FALSE
  rand_slope = FALSE

  # Check if it's a linear or mixed effects model
  if (class(model)[1] == "lmerModLmerTest") {
    cat("This is a mixed effects linear model\n")
    is.lmer = TRUE
  } else if (class(model)[1] == "lm") {
    cat("This is a linear model\n")
  } else {
    stop("The input model is not a linear model or a mixed effects linear model.")
  }

  # Create diagnostic plots
  plot_list <- list()
  model.data = data.frame(cbind("fitted" = fitted(model), "residuals" = resid(model),
                                "hatVs" = hatvalues(model)))

  # Residuals vs. Fitted plot
  plot_list$resid_fit <- ggplot(model.data, aes(fitted(model), resid(model))) +
    geom_point() +

```



```

ggtitle("Residuals vs. Fitted") +
geom_smooth(se = FALSE, color = "red") +
xlab("Fitted values") +
ylab("Residuals")

# Q-Q plot
plot_list$qq <- ggplot(model.data, aes(sample = resid(model))) +
  geom_qq() +
  geom_qq_line(color = "red") +
  ggtitle("Q-Q Plot") +
  xlab("Theoretical Quantiles") +
  ylab("Sample Quantiles")

# Scale-location plot
plot_list$scale_loc <- ggplot(model.data, aes(fitted(model), sqrt(abs(resid(model))))) +
  geom_point() +
  geom_smooth(se = FALSE, color = "red") +
  ggtitle("Scale-Location Plot") +
  xlab("Fitted values") +
  ylab("sqrt(|Standardized residuals|)")

# Residuals vs. Leverage plot
plot_list$resid_lev <- ggplot(model.data, aes(hatvalues(model), resid(model))) +
  geom_point() +
  ggtitle("Residuals vs. Leverage") +
  geom_smooth(se = FALSE, color = "red") +
  xlab("Leverage") +
  ylab("Residuals")

if (is.lmer){
  # only do qqplots for random each random effect (ranef)
  # get ranef residuals
  df.ranef = data.frame(ranef(model))

  ran_eff_plots = list()
  for (ran_eff in unique(df.ranef$term)) {
    chunk.data = df.ranef[df.ranef$term == ran_eff, ]
    # check for outliers
    chunk.data$conoval.outliers = ID.outliers(chunk.data$conoval, 3)
    chunk.data$conosd.outliers = ID.outliers(chunk.data$conosd, 3)

    # Normal distribution of random effect values
    plot_name = paste0(ran_eff, ".VarNorm")
    ran_eff_plots[[plot_name]] = ggplot(chunk.data, aes(x = conoval, y = conosd, color =
conosd.outliers)) +
      geom_text(aes(label = ifelse(conosd.outliers, grp, NA)),
        na.rm = TRUE, nudge_x = (sum(abs(range(chunk.data$conoval)))/100)*2) +
      geom_point() +
      scale_color_manual(values = c("black", "red")) +
      ggtitle("RanEff Variability") +
      geom_smooth(se = FALSE, color = "red") +
      xlab("Conditional Effects") +
      ylab("Standard Deviation") +
      theme(legend.position = "none")

    plot_name = paste0(ran_eff, ".ValNorm")
    # Normal distribution of random effect variability
    ran_eff_plots[[plot_name]] = ggplot(chunk.data, aes(sample = conoval, color =
conoval.outliers)) +
      stat_qq() +
      scale_color_manual(values = c("black", "red")) +
      geom_qq_line(color = "red") +
      ggtitle(paste0("Q-Q RanEff Values ", ran_eff)) +
      xlab("Theoretical Quantiles") +
      ylab("Sample Quantiles") +
      theme(legend.position = "none")
  }
}

```

```

# DW test for autocorrelation
# between 1.5 and 2.5 = ideal. below and above = risk of autocorrelation
Results$DurbinWatson = durbinWatsonTest(resid(model))

# Save plots
if (!is.lmer){
  arr_plot = ggarrange(plot_list$resid_fit,
                        plot_list$qq,
                        plot_list$scale_loc,
                        plot_list$resid_lev)
}else{
  # find out how many plots need to be there
  arr_plot = ggarrange(plotlist = c(plot_list, ran_eff_plots),nrow = 4, ncol = 2)
}
annotate_figure(arr_plot, top = text_grob((paste0("Assumptions for Model: \n",
model.fullname, "\n",
                                                "Durbin-Watson: ",
as.character(Results$DurbinWatson))),
                                                face = "bold", size = 14))

arr_plot
# Save plots
ggsave(filename = paste0(save_path, "-AssumptionsPlot-", model.name, ".pdf"), width = 8,
height = 8)

Results$Plots = plot_list

return(Results)
}

# Function to convert formula to string
formula_to_string <- function(formula) {
  # Convert formula items to strings
  formula_strings <- lapply(as.list(formula), deparse)
  # Return the formula as a string
  paste(c(formula_strings[[2]], formula_strings[[1]], formula_strings[[3]]), collapse = "
")
}

# Function to convert string to formula
string_to_formula <- function(string) {
  # Convert the string to a formula
  as.formula(string)
}

# do a lmer model with 4 cores
fourcore_lmer = function(formula, data){
  # Initialize a parallel backend with 4 cores
  cl <- makeCluster(4)

  # Register the parallel backend
  registerDoParallel(cl)

  # Set the parallel backend as the default
  options(mc.cores = 4)

  # Specify the parallel processing control object
  ctrl <- lmerControl(calc.derivs = FALSE)

  # Fit a mixed effects model with random intercept and random slope using parallel
processing
  model <- lmer(formula, data=data, control = ctrl)

  # Print the model summary
  summary(model)

  # Stop the parallel backend
  stopCluster(cl)
}

```

```

    return(model)
}

# Function to perform linear regression with random effects and two variables
add2model_regression_function <- function(data, formula, variable1, variable2) {

  # Check assumptions for linear regression
  assumption_results <- list()
  assumption_results$normality <- shapiro.test(resid(lmer(formula, data=data)))$p.value <
0.05
  assumption_results$homoscedasticity_variable1 <- bartlett.test(resid(lmer(formula,
data=data)) ~ data[[variable1]])$p.value < 0.05
  assumption_results$homoscedasticity_variable2 <- bartlett.test(resid(lmer(formula,
data=data)) ~ data[[variable2]])$p.value < 0.05

  # Linear model with no added effects
  model1 <- lmer(formula, data=data)

  # Linear model with 1 variable in
  V1_formula = strsplit(formula_to_string(formula), split = "~ 1")
  V1_formula = as.formula(paste(V1_formula[[1]][1], '~', variable1,
                                V1_formula[[1]][2], split = ""))
  modelV1 <- lmer(V1_formula, data=data)

  # Linear model with other 1 variable in
  V2_formula = strsplit(formula_to_string(formula), split = "~ 1")
  V2_formula = as.formula(paste(V2_formula[[1]][1], '~', variable2,
                                V2_formula[[1]][2], split = ""))
  modelV2 <- lmer(V2_formula, data=data)

  # Linear model with both variable in
  model_comp_noint_formula = strsplit(formula_to_string(formula), split = "~ 1")
  model_comp_noint_formula = as.formula(paste(model_comp_noint_formula[[1]][1], '~',
variable2,
                                ' + ', variable1,
                                model_comp_noint_formula[[1]][2], split = ""))
  model_complete_noint <- lmer(model_comp_noint_formula, data=data)

  # Linear model with both variable in
  model_comp_int_formula = strsplit(formula_to_string(formula), split = "~ 1")
  model_comp_int_formula = as.formula(paste(model_comp_int_formula[[1]][1], '~', variable2,
                                ' * ', variable1,
                                model_comp_int_formula[[1]][2], split = ""))
  model_complete_int <- lmer(model_comp_int_formula, data=data)

  # Model summaries
  model_summaries <- list()
  model_summaries$model1_summary <- summary(model1)
  model_summaries$model2_summary <- summary(modelV1)
  model_summaries$model3_summary <- summary(modelV2)
  model_summaries$model4_summary <- summary(model_complete_noint)
  model_summaries$model5_summary <- summary(model_complete_int)

  # Model comparison
  model_comparison <- list()
  model_comparison$model1_vs_model2 <- try(anova(model1, modelV1, modelV2,
                                model_complete_noint, model_complete_int),
silent = TRUE)

  # check if model3 built successfully
  if (inherits(model_comparison$model1_vs_model2, "try-error")) {
    warning("Model comparison did not work due to the error:",
model_comparison$model1_vs_model2[1])
  } else {
    # Print the model summary
  }
}

```

```

# Return results
results <- list()
results$assumption_results <- assumption_results
results$model_summaries <- model_summaries
results$model_comparison <- model_comparison
results$model1 = model1
results$modelV1 = modelV1
results$modelV2 = modelV2
results$model_comp_noint_formula = model_comp_noint_formula
results$model_comp_int_formula = model_comp_int_formula
# results$recommendation <- recommendation

return(results)
}

# Function to perform linear regression with random effects
nullmodel_regression_function <- function(data, formula, variable) {

  # Check assumptions for linear regression
  assumption_results <- list()
  assumption_results$normality <- shapiro.test(resid(lm(formula, data=data)))$p.value <
0.05
  assumption_results$homoscedasticity <- bartlett.test(resid(lm(formula, data=data)) ~
data[[variable]])$p.value < 0.05

  # Linear model with no random effects
  model1 <- lm(formula, data=data)

  # Linear model with random intercept
  random_int_formula = as.formula(paste(formula_to_string(formula), " + (1|", variable,
  ")", sep = ""))
  model2 <- lmer(random_int_formula, data=data)

  # Linear model with random intercept and slope
  random_slo_formula = as.formula(paste(formula_to_string(formula), " + (", variable, "|",
variable, ")", sep = ""))
  # envelop model3 in a try block
  model3 <- try(fourcore_lmer(random_slo_formula, data=data), silent = TRUE)

  # check if model3 built successfully
  if (inherits(model3, "try-error")) {
    warning("Model 3 (random intercept and slope) could not be fit due to the error:",
model3[1])
    model3_summary <- summary(model2)
    model3 = model2
    warning("Model 3 replaced with Model 2 so this function does not stop working")
  } else {
    # Print the model summary
    model3_summary <- summary(model3)
  }

  # Model summaries
  model_summaries <- list()
  model_summaries$model1_summary <- summary(model1)
  model_summaries$model2_summary <- summary(model2)
  model_summaries$model3_summary <- model3_summary

  # Model comparison
  model_comparison <- list()
  model_comparison$model1_vs_model2 <- anova(model2, model1, model3)

  # Recommendation
  if (AIC(model1) < AIC(model2) && AIC(model1) < AIC(model3)) {
    recommendation <- "Model 1 (no random effects) is recommended."
  } else if (AIC(model2) < AIC(model3)) {
    recommendation <- "Model 2 (random intercept) is recommended."
  } else {
    recommendation <- "Model 3 (random intercept and slope) is recommended."
  }
}

```

```

}

# Return results
results <- list()
results$assumption_results <- assumption_results
results$model_summaries <- model_summaries
results$model_comparison <- model_comparison
results$recommendation <- recommendation
results$models <- c(formula_to_string(formula),
                    formula_to_string(random_int_formula),
                    formula_to_string(random_slo_formula))

return(results)
}

# MEP functions -----
ExciseUpDown <- function(x) {
  # Convert the input to a character vector
  x <- as.character(x)

  # Replace "up" and "down" with an empty string and also take out the block number
  x <- gsub("(up|down)\\.\\.\\.\\d+", "", x)

  # Convert the resulting character vector back to a factor
  x <- as.factor(x)

  # Return the modified factor
  return(x)
}

# this function makes a paired t.test between the up and down conditions
MEP_Test_Block <- function(data, sess, block, UID = "UID", comparison1 = "up", comparison2
= "down",
                           testname = "paired.ttest",
                           directionality = "greater"){
  # create results list
  result = list()

  # Create Dataframes of UID and the respective MEP block, session, and condition
  comp1.up.data = data[(data[, "plot.sessions"] == sess &
                        as.character(data[, "plot.blocks"]) == block &
                        data[, "conditions"] == comparison1),
                      c("actual_MEP", UID)]
  comp2.down.data = data[(data[, "plot.sessions"] == sess &
                          as.character(data[, "plot.blocks"]) == block &
                          data[, "conditions"] == comparison2),
                        c("actual_MEP", UID)]

  # Take out the "up" and "down" part of UID as it will mess up
  comp1.up.data$UID = ExciseUpDown(comp1.up.data$UID)
  colnames(comp1.up.data) = c("Actual_MEP_Up", "UID")
  comp2.down.data$UID = ExciseUpDown(comp2.down.data$UID)
  colnames(comp2.down.data) = c("Actual_MEP_Down", "UID")

  # Make DF paired
  paired.data = MakePaired(comp1.up.data, comp2.down.data, UID)
  # Compile Results
  result$data$excluded = paired.data$excluded
  result$data$data = paired.data$result

  test = PairedTest(paired.data$result$Actual_MEP_Up, paired.data$result$Actual_MEP_Down,
                    testName = testname,
                    directionality = "greater")

```

```

# Compile Results
result$test$name = test$name
result$test$ShapiroWilk_Norm = test$ShapiroWilk_Norm
result$test$up_summ = test$x_summ
result$test$down_summ = test$y_summ
result$test$test = test$test

return(result)
}

MEP_Test_Block_Save <- function(Data, SaveName, SaveDir = './'){
  write.csv(Data$data$excluded, file = paste(SaveDir, SaveName, '_excluded.csv', sep = ""))
  write.csv(Data$data$data, file = paste(SaveDir, SaveName, '_data.csv', sep = ""))
  write(capture.output(Data$test), file = paste(SaveDir, SaveName, '_test.txt', sep = ""))
}

MEP_learner_classification <- function(Data, MEP_Column_Name){
  # 1. For each value of the variable PID do the following
  for (subjID in unique(Data$PID)) {

    # 2. take a subsection of the data that has the PID value
    current.df = Data[Data$PID == subjID, ]

    # 3. build a linear model predicting the column-name variable using two variables
    called condition and mep.nr.cont
    linmod = lmer(MEP_Column_Name ~ conditions * mep.nr.train, data = Data)

    # 4. if the variable condition or the interaction between mep.nr.cont and condition is
    significant classify this PID as "TRUE" in a column called "learner" in the original
    dataframe
    # 5. save the model and the output
  }
  # 6. Return the dataframe with the additional column of "learner"
}

Save_lm_summ = function(Linear_Model, File_Name = "Path/Linear"){
  save(Linear_Model, file = paste0(File_Name, "-model.rda"))
  write(capture.output(summary(Linear_Model)), file = paste0(File_Name, "-output.txt"))
}

# StabDist Functions -----

Check_For_Outliers <- function(chunk.data.x = NULL, chunk.data.y, pdf.path.name,
outlier_sd_range = 3){
  # look at outliers
  # chunk.data = df.StabDist$Distance.Rest.UP
  results = list()
  chunk.3Xsd = outlier_sd_range*sd(chunk.data.y)
  chunk.mean = mean(chunk.data.y)
  results$summary = summary(chunk.data.y)
  print(results$summary)

  # Check outliers and return them
  results$outliers = ID.outliers(chunk.data.y, outlier_sd_range)

  # Plot outliers and save
  pdf(file = pdf.path.name)
  if (is.null(chunk.data.x)){
    plot(#x = chunk.data.x,
         chunk.data.y,
         ylim = c(min(chunk.data.y)-0.5*chunk.3Xsd,
                   max(chunk.data.y)+0.5*chunk.3Xsd),
         col = ifelse(results$outliers, "red", "black"))
  } else {
    plot(x = chunk.data.x,
         y = chunk.data.y,
         ylim = c(min(chunk.data.y)-0.5*chunk.3Xsd,

```

```

        max(chunk.data.y)+0.5*chunk.3Xsd),
        col = ifelse(results$outliers, "red", "black"))
    }
    abline(h = chunk.mean+chunk.3Xsd, col = "blue")
    abline(h = chunk.mean, col = "red")
    abline(h = chunk.mean-chunk.3Xsd, col = "blue")
    dev.off()
    return(results)
}

BCI_Achiever <- function(dataframe, PID = "PID", CA = "CA"){
  dataframe$Achiever.kss = NA
  dataframe$Achiever.m = NA
  dataframe$t.zero = NA
  dataframe$t.lastS = NA
  dataframe$t.Slope = NA
  dataframe$FirstVLast = NA

  for (pid in unique(dataframe[[PID]])) {
    chunk.data = dataframe[dataframe$PID == pid,]
    # build model
    achiever.m = lm(CA ~ Continuous.Block, data = chunk.data)
    ## with KSS
    achiever.kss = lm(CA ~ Continuous.Block + KSS, data = chunk.data)

    # check if interaction is significant
    achiever.m.pval = summary(achiever.m)$coefficients["Continuous.Block", "Pr(>|t|)"]
    # KSS
    achiever.kss.pval = summary(achiever.kss)$coefficients["Continuous.Block", "Pr(>|t|)"]
    # first sess
    first_sess = ifelse(chunk.data$Group[1] == "TMS", 3, 1)

    # t.test different from zero
    t.zero = t.test(c(chunk.data$CA-50))$p.value
    # t.test different from zero last sess
    t.lastS = t.test(c(chunk.data$CA[chunk.data$Sess == 5]-50))$p.value
    # t.test different from zero slope
    slope = chunk.data$CA[chunk.data$Sess == 5] - chunk.data$CA[chunk.data$Sess ==
first_sess]
    t.Slope = t.test(slope)$p.value
    # t.test first vs last session
    firstvlastsess = PairedTest(chunk.data$CA[chunk.data$Sess == first_sess],
chunk.data$CA[chunk.data$Sess == 5],
                                testName = "CA First vs Last Session", directionality =
"less")
    t.fvla = firstvlastsess$test$p.value

    # categorise participant
    dataframe$Achiever.kss[dataframe$PID == pid] = achiever.kss.pval
    dataframe$Achiever.m[dataframe$PID == pid] = achiever.m.pval
    dataframe$t.zero[dataframe$PID == pid] = t.zero
    dataframe$t.lastS[dataframe$PID == pid] = t.lastS
    dataframe$t.Slope[dataframe$PID == pid] = t.Slope
    dataframe$FirstVLast[dataframe$PID == pid] = t.fvla
  }
  return(dataframe)
}

# Date Functions -----

# Recursive function to search for a file and retrieve creation date
search_file_recursive <- function(folder_path, target_file, disc_inpath = "disc") {
  result = list()

  files <- list.files(folder_path, full.names = TRUE,
                      pattern = target_file, recursive = TRUE)

```

```

for (file in files) {
  # Check if "disc" is present in the path
  if (!grepl(disc_inpath, file, ignore.case = TRUE)) {
    # Retrieve creation date
    creation_date <- format(as.POSIXct(file.info(file)$mtime, origin="1970-01-01"))
    message(paste("File '", target_file, "' found. Created on: ", creation_date))

    # Extract numbers from the file name
    nf_match <- regmatches(file, regexpr("nf\\d+", file, ignore.case = TRUE))
    sess_match <- regmatches(file, regexpr("sess\\d+", file, ignore.case = TRUE))
    nf_number <- as.numeric(gsub("nf", "", nf_match))
    sess_number <- as.numeric(gsub("sess", "", sess_match))
    result_ind = length(result$PID)+1
    result$PID[result_ind] = nf_number
    result$sess[result_ind] = sess_number
    result$CreationDate[result_ind] = creation_date
    result$Path[result_ind] = file
  }
}
return(result)
}

# Publication Functions -----

# create a APA conform excel table form coefficientents
format_coeff_table_apa = function(model,
                                  select.columns = c("Estimate", "t value", "Pr(>|t|)"),
                                  save.path.name = "../coefficient-table.xlsx",
                                  replace.variables = NA,
                                  replacement.variables = NA,
                                  save.as.excel = TRUE){

  require(openxlsx)
  df.result = summary(model)$coefficients

  # Select Columns
  df.result = df.result[, c("Estimate", "t value", "Pr(>|t|)")]

  ## Format Table -----
  # round (signif/round combo?)
  # slightly different format for t values and betas than for p values
  df.result.betas = apply(df.result, 2,
                           function(x) ifelse(abs(x) > 1, round(x, 2), signif(x, 2)))
  df.result.pvals = apply(df.result, 2,
                           function(x) ifelse(abs(x) > 1, round(x, 2), ifelse(abs(x) >
0.001, signif(x, 2), "0.001 >")))
  df.result = cbind(df.result.betas[, c("Estimate", "t value")],
                    df.result.pvals[, c("Pr(>|t|)")])

  # browser()
  # replace variable names
  if (!sum(is.na(replace.variables))) {
    for (to.replace in 1:length(replace.variables)) {
      rownames(df.result) <- gsub(replace.variables[to.replace],
replacement.variables[to.replace], rownames(df.result))
    }
  }

  # replace text strings
  colnames(df.result) <- c("Beta", "t-Value", "p-Value")
  rownames(df.result) <- gsub("\\.", " : ", rownames(df.result))

  df.result = data.frame(df.result)

```



```

if (save.as.excel) {
  ## Save -----
  # save as preformatted as possible
  # Create a new workbook
  wb <- createWorkbook()

  # Add a worksheet
  addWorksheet(wb, "MySheet")

  # set font type
  modifyBaseFont(wb, fontSize = 10, fontColour = "black", fontName = "Times New Roman")

  # Write the data frame to the worksheet
  writeData(wb, "MySheet", df.result, rowNames = TRUE)

  # Add a border to the first row
  addStyle(wb, "MySheet",
    style = createStyle(border = "Bottom"),
    rows = 1,
    cols = 1:(ncol(df.result) + 1))

  # Add a border to the last row
  addStyle(wb, "MySheet",
    style = createStyle(border = "Bottom"),
    rows = nrow(df.result)+1,
    cols = 1:(ncol(df.result) + 1))

  # set auto width for first col
  setColWidths(wb, "MySheet", cols = 1, widths = "auto")

  # center all cells but variable rows
  size.result = dim(df.result)
  centerStyle <- createStyle(halign = "center")
  # noneed mergeCells(wb, "TestSheet", cols=1:length(df), rows=1)
  addStyle(wb, "MySheet", centerStyle, rows = 1:(size.result[1]+2), cols =
2:(size.result[2]+1), gridExpand = TRUE, stack = TRUE)

  # Save the workbook
  # browser()
  name.workbook = paste0(save.path.name, "coeff-table.xlsx")
  saveWorkbook(wb, name.workbook, overwrite = TRUE)
  # browser()
}
return(df.result)
}

```

10.4.2 MEP Check Script (MATLAB)

This script displays all MEPs on the same scale and allows the user to categorise them as ‘Ok’, ‘Strange’, or ‘Bad’. Any MEPs not categorised as ‘Ok’ were excluded from analysis.

```
% This script looks at all MEPs and cats them as ok/strange/bad
% Colin 2023/02/22

clear
%% initialisation

% guiding the script (when you don't want to look at all meps
% starting from?
start_ind_pp = 1;
start_ind_b = 1;
% want to look at prepost meps?
prepost_guide = 1;
% want to look at block meps?
block_guide = 0;
% append?
append_block = 0;
append_pp = 0;

% get all meps
path = '';
% addpath();

cd(path);

save_path = '230221 Analysis/Processed Data/MEP Data/MEP_Cat';

prepost_meps = dir("nf*/sess*/*-mep/*-mep*.mat");
block_meps = dir("nf*/sess*/block*/*-mep*.mat");

responses= {};

%% continue or leave indices?
if block_guide
    % check if a file exists, if it does read it in
    if exist("230221 Analysis/Processed Data/MEP
Data/MEP_Cat/Block_MEP_Categorisation.csv")
        % read in file
        block_mep_cat = readtable("230221 Analysis/Processed Data/MEP
Data/MEP_Cat/Block_MEP_Categorisation.xlsx")

        % ask if continue indexing or not
        choice_continue = questdlg('Do you want to continue where you left off or leave
indices as defined?', ...
        'Continue or Leave indices?', ...
        'Continue', 'Leave Indices', 'Continue');
        switch choice_continue
            case 'Continue'
                start_ind_b = size(block_mep_cat, 1) + 1
                append_block = 1;
            case 'Leave Indices'
        end
    end
end

if prepost_guide
    % check if a file exists, if it does read it in
    if exist("230221 Analysis/Processed Data/MEP
Data/MEP_Cat/Prepost_MEP_Categorisation.csv")
        % read in file
        prepost_mep_cat = readtable("230221 Analysis/Processed Data/MEP
Data/MEP_Cat/Prepost_MEP_Categorisation.xlsx")

        % ask if continue indexing or not
        choice_continue = questdlg('Do you want to continue where you left off or leave
indices as defined?', ...
```

```

        'Continue or Leave indices?', ...
        'Continue', 'Leave Indices', 'Continue');
    switch choice_continue
        case 'Continue'
            start_ind_pp = size(prepost_mep_cat, 1) + 1;
            responses = prepost_mep_cat.Cat';
        case 'Leave Indices'
    end
end
end
end

%% loop over all prepost meps meps
if prepost_guide
    for i = start_ind_pp:size(prepost_meps,1)
        clear feedbackData
        load([prepost_meps(i).folder '/' prepost_meps(i).name]);
        TMS_onset = get_TMS_onset(feedbackData(:,1));
        plot_data = feedbackData((TMS_onset-200):(TMS_onset+700), 1:5);
        plot_data(:,1) = plot_data(:,1)/4;
        figure('units','normalized','outerposition',[0 0 1 1])
        plot(plot_data);
        ylim([-1,1])
        pause(0.1)

        % Cat figure
        answer = questdlg('How do you Categorise this MEP', ...
            'MEP Category', ...
            'Normal','Not Normal','Go Back or Stop', 'Normal');
        % Handle response
        switch answer
            case 'Normal'
                responses{i} = 'Normal';

            case 'Not Normal'
                responses{i} = 'Not Normal';
            case 'Go Back or Stop'
                choice = questdlg('Go back or stop?', ...
                    'MEP Category', ...
                    'Go Back', 'Stop', 'Go Back')
                switch choice
                    case 'Go Back'
                        % Cat figure
                        go_back_answer = questdlg('The Previous Figure was:', ...
                            'MEP Category', ...
                            'Normal','Not Normal', 'Normal');
                        switch go_back_answer
                            case 'Normal'
                                responses{i-1} = 'Normal';

                                case 'Not Normal'
                                    responses{i-1} = 'Not Normal';
                                end
                        % Cat figure
                        answer = questdlg('How do you Categorise this MEP', ...
                            'MEP Category', ...
                            'Normal','Not Normal', 'Normal');
                        % Handle response
                        switch answer
                            case 'Normal'
                                responses{i} = 'Normal';

                                case 'Not Normal'
                                    responses{i} = 'Not Normal';
                                end
                        case 'Stop'
                            i
                            break
                        end
                    end

                end

            end

        % Close figure
        pause(0.05)
        close;
    end
    folders = {prepost_meps.folder}';
    names = {prepost_meps.name}';
    % how far we got
end

```

```

nr_end = length(responses);
% make table as far as we got:
prepost_table = table(folders(1:nr_end), names(1:nr_end), responses', ...
    VariableNames={'Folder', 'Name', 'Cat'})

writetable(prepost_table, [save_path, '/Prepost_MEP_Categorisation.xlsx'])
writetable(prepost_table, [save_path, '/Prepost_MEP_Categorisation.csv'])
end

%% loop over all prepost meps meps
if block_guide
    for i = start_ind_b:size(block_meps,1)
        clear feedbackData
        load([block_meps(i).folder '/' block_meps(i).name]);
        TMS_onset = get_TMS_onset(feedbackData(:,1));
        plot_data = feedbackData((TMS_onset-200):(TMS_onset+700), 1:3);
        plot_data(:,1) = plot_data(:,1)/4;
        figure('units','normalized','outerposition',[0 0 1 1])
        plot(plot_data);
        ylim([-1,1])
        pause(0.1)

        % Cat figure
        answer = questdlg('How do you Categorise this MEP', ...
            'MEP Category', ...
            'Normal','Not Normal','Go Back or Stop', 'Normal');
        % Handle response
        switch answer
            case 'Normal'
                responses{i} = 'Normal';

            case 'Not Normal'
                responses{i} = 'Not Normal';

            case 'Go Back or Stop'
                choice = questdlg('Go back or stop?', ...
                    'MEP Category', ...
                    'Go Back', 'Stop', 'Go Back')
                switch choice
                    case 'Go Back'
                        % Cat figure
                        go_back_answer = questdlg('The Previous Figure was:', ...
                            'MEP Category', ...
                            'Normal','Not Normal', 'Normal');
                        switch go_back_answer
                            case 'Normal'
                                responses{i-1} = 'Normal';

                            case 'Not Normal'
                                responses{i-1} = 'Not Normal';
                        end
                        % Cat figure
                        answer = questdlg('How do you Categorise this MEP', ...
                            'MEP Category', ...
                            'Normal','Not Normal', 'Normal');
                        % Handle response
                        switch answer
                            case 'Normal'
                                responses{i} = 'Normal';

                            case 'Not Normal'
                                responses{i} = 'Not Normal';
                        end
                    case 'Stop'
                        i
                        break
                end
            end
        end
        % Close figure
        close;
        pause(0.05)
    end
    folders = {block_meps.folder}';
    names = {block_meps.name}';
    % how far we got
    nr_end = length(responses);
    %make table as far as we got:

```

```

        prepost_table = table(folders(start_ind_pp:nr_end), names(start_ind_pp:nr_end),
responses', ...
        VariableNames={'Folder', 'Name', 'Cat'})

        writetable(prepost_table, [save_path, '/Block_MEP_Categorisation.xlsx'])
        writetable(prepost_table, [save_path, '/Block_MEP_Categorisation.csv'])
end

```

10.4.3 EEG Signal Analysis (MATLAB)

This script handles the split into different powerbands.

```
% Signal Analysis Script
close all
clear all
clc

%% Setting up the structure

% load better colormaps
load('')
% load better colormaps
load('')
load('')
load('')

% work dir
work_dir = '';

% save dir:
save_dir = '';

% add EEGLab
addpath(''); %eeglab

% add mortlet power
addpath('')

subj_list= {''}';
condition_list= {'UP', 'DN'};

%% Frequencies
% Frequencies we want to have a look at it will be like this min <= freq < max
HighDelta_min_freq = 1;
HighDelta_max_freq = 5;

Theta_min_freq = 5;
Theta_max_freq = 8;

LowAlpha_min_freq = 8;
LowAlpha_max_freq = 11;

HighAlpha_min_freq = 11;
HighAlpha_max_freq = 14;

LowBeta_min_freq = 14;
LowBeta_max_freq = 22;

HighBeta_min_freq = 22;
HighBeta_max_freq = 31;

LowGamma_min_freq = 31;
LowGamma_max_freq = 45;

Global_min_freq = 1;
Global_max_freq = 45;

OpenVibe_min_freq = 8;
OpenVibe_max_freq = 30;

%% Timings
EEG_srate = 250;
EEG_size = 7 * EEG_srate;
% so we get whole data period
EEG_snip = floor(0.375 * EEG_srate);

Baseline_idx = [1 2*EEG_srate];
Condition_idx = [((EEG_size - EEG_snip) - 3 * EEG_srate) ...
    (EEG_size - EEG_snip)];

%just for now
Condition_idx = Condition_idx - (Condition_idx(2) - 1188);
```

```

%% Initiation

analysis_diagnostics = {};

% Establish empty matrices to later hold
HighDelta_data_percchange = [];
Theta_data_percchange = [];
LowAlpha_data_percchange = [];
HighAlpha_data_percchange = [];
LowBeta_data_percchange = [];
HighBeta_data_percchange = [];
LowGamma_data_percchange = [];
Global_data_percchange = [];
OpenVibe_data_percchange = [];

HighDelta_data_relpower = [];
Theta_data_relpower = [];
LowAlpha_data_relpower = [];
HighAlpha_data_relpower = [];
LowBeta_data_relpower = [];
HighBeta_data_relpower = [];
LowGamma_data_relpower = [];
Global_data_relpower = [];
OpenVibe_data_relpower = [];

cd(work_dir);

% open EEGLab
[ALLEEG EEG CURRENTSET ALLCOM] = eeglab;

%% Reading in Files
% we now have new folder structure with all files in one folder!
current_filepath= [work_dir '']

cd(current_filepath) %All the sessions and blocks are in here. Treat every file the same
for now- can organise into ses and blocks afterwards
contents = dir([current_filepath filesep '*.set']);
dirFlags = ~[contents.isdir] & ~strcmp({contents.name},'.') &
~strcmp({contents.name}, '..');
files=contents(dirFlags); %list only the files that are not directories

subj_data=[]; %empty vector for this subj

for k=1:length(files) %for every file in the folder, apply the analysis and save results
    % I'll try to have it count backwards to see why it struggles towards
    % the end
    try
        current_idx = length(files)+1 - k
        current_filename=files(current_idx).name
        % Extract the session and block numbers
        pid = regexp(current_filename, 'sub-(\d+)', 'tokens');
        pid = str2double(pid{1}{1});

        ses = regexp(current_filename, 'sess-?(\d+)', 'tokens');
        ses = str2double(ses{1}{1});

        block = regexp(current_filename, 'block(\d+)|b(\d+)\.', 'tokens');
        block = str2double(block{1}{1});

        freqs=(1:1:45)';
        Fs=250; %For pwelch to work, this vector needs to begin at 0 and go to 125 (find this
by trial and error, depends on sampling freq and appears to always be half)

        %% Split Dataset into Up and Dn
        for n = 1:length(condition_list)
            %% Load file
            % To make sure we don't paste the wrong data
            HighDelta_mean_percchange = [];
            Theta_mean_percchange = [];
            LowAlpha_mean_percchange = [];
            HighAlpha_mean_percchange = [];
            LowBeta_mean_percchange = [];
            HighBeta_mean_percchange = [];
            LowGamma_mean_percchange = [];
            Global_mean_percchange = [];
            Openvibe_mean_percchange = [];

```

```

HighDelta_mean_relpower = [];
Theta_mean_relpower = [];
LowAlpha_mean_relpower = [];
HighAlpha_mean_relpower = [];
LowBeta_mean_relpower = [];
HighBeta_mean_relpower = [];
LowGamma_mean_relpower = [];
Global_mean_relpower = [];
Openvibe_mean_relpower = [];

%Below is to import the file to be processed I did not find
%a way to import create a second dataset with the file (for
%up and down) so we import it twice :/
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 1,'gui','off'); %

EEG = pop_loadset('filename', current_filename, 'filepath',
current_filepath); %load the data .set file
EEG = eeg_checkset( EEG );

% list of markers:
marker_list = unique({EEG.event.type});
marker_right = strcmp(unique(marker_list), 'OVTK_GDF_Right');
marker_left = strcmp(unique(marker_list), 'OVTK_GDF_Left');

marker_list = {marker_list{marker_right}, marker_list{marker_left}};

% Select Epochs based on Marker
EEG = pop_selectevent( EEG, 'type',
marker_list(n),'deleteevents','off','deleteepochs','on','invertepochs','off');
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 2,'gui','off');

%% Do PercChange
% new mortlet power way:
power_data = Compute_Mortlet_csi(EEG);

% Split into percchange metric and
power_data_percchange =
zeros(size(power_data(:,Condition_idx(1):Condition_idx(2),:,:)));
freqs_idx = size(power_data, 1);
time_idx = size(power_data, 2);
epochs_idx = size(power_data, 3);
electrodes_idx = size(power_data, 4);

% every Epoch
for epoch = 1:epochs_idx
    % every Freq
    for frequency = 1:freqs_idx
        %for every electrode
        for electrode = 1:electrodes_idx
            % get mean
            ep_freq_elec_mean = mean(power_data(frequency,
Baseline_idx(1):Baseline_idx(2), epoch, electrode));
            % do normalisation into percentage change
            power_data_percchange(frequency, :, epoch, electrode) = ...
100 * ((power_data(frequency, Condition_idx(1):Condition_idx(2),
epoch, electrode) - ep_freq_elec_mean)/ep_freq_elec_mean);
        end
    end
end

%% Also dbnorm
% Split into percchange metric and
power_data_dbnorm=
zeros(size(power_data(:,Condition_idx(1):Condition_idx(2),:,:)));

% every Epoch
for epoch = 1:epochs_idx
    % every Freq
    for frequency = 1:freqs_idx
        %for every electrode
        for electrode = 1:electrodes_idx
            % get mean
            ep_freq_elec_mean = mean(power_data(frequency,
Baseline_idx(1):Baseline_idx(2), epoch, electrode));
            % do normalisation into percentage change

```



```

        power_data_dbnorm(frequency, :, epoch, electrode) = ...
            10 * log10(power_data(frequency, Condition_idx(1):Condition_idx(2),
epoch, electrode)/ep_freq_elec_mean);
    end
end
end
time_idx = size(power_data_dbnorm, 2);

%% calc relpower from dbnorm
% Split into percchange metric and
power_data_relpow= zeros(size(power_data_dbnorm));

for epoch = 1:epochs_idx
    % every Freq
    for frequency = 1:freqs_idx
        %for every electrode
        for electrode = 1:electrodes_idx
            % for every timepoint
            % for timepoint = 1:time_idx
            % calc total power
            total_pow = sum(abs(power_data_dbnorm(:, :, epoch, electrode)));
            % calculate relpower
            power_data_relpow(frequency, :, epoch, electrode) = ...
                100 * rdivide(abs(power_data_dbnorm(frequency, :, epoch,
electrode)),total_pow);
            % end
        end
    end
end

%% put together

%%Average the spectrum across all epochs at the condition times:
Mean_freq_chan_percchange = mean(mean(power_data_percchange, 3), 2);
Mean_freq_chan_percchange = squeeze(Mean_freq_chan_percchange);

Mean_freq_chan_relpower = mean(mean(power_data_relpow, 3), 2);
Mean_freq_chan_relpower = squeeze(Mean_freq_chan_relpower);

%% BETA/ALPHA/WHATEVER YOU WANT TO SUMMARISE
%find only the rows in the spectrum with values at the freqs
%you want
HighDelta_vect_F = find(freqs >= HighDelta_min_freq & freqs <
HighDelta_max_freq);
Theta_vect_F = find(freqs >= Theta_min_freq & freqs < Theta_max_freq);
LowAlpha_vect_F = find(freqs >= LowAlpha_min_freq & freqs <
LowAlpha_max_freq);
HighAlpha_vect_F = find(freqs >= HighAlpha_min_freq & freqs <
HighAlpha_max_freq);
LowBeta_vect_F = find(freqs >= LowBeta_min_freq & freqs <
LowBeta_max_freq);
HighBeta_vect_F = find(freqs >= HighBeta_min_freq & freqs <
HighBeta_max_freq);
LowGamma_vect_F = find(freqs >= LowGamma_min_freq & freqs <
LowGamma_max_freq);
Global_vect_F = find(freqs >= Global_min_freq & freqs < Global_max_freq);
OpenVibe_vect_F = find(freqs >= OpenVibe_min_freq & freqs <
OpenVibe_max_freq);

HighDelta_spectrum_relpower = Mean_freq_chan_relpower(HighDelta_vect_F, :);
Theta_spectrum_relpower = Mean_freq_chan_relpower(Theta_vect_F, :);
LowAlpha_spectrum_relpower = Mean_freq_chan_relpower(LowAlpha_vect_F, :);
HighAlpha_spectrum_relpower = Mean_freq_chan_relpower(HighAlpha_vect_F, :);
LowBeta_spectrum_relpower = Mean_freq_chan_relpower(LowBeta_vect_F, :);
HighBeta_spectrum_relpower = Mean_freq_chan_relpower(HighBeta_vect_F, :);
LowGamma_spectrum_relpower = Mean_freq_chan_relpower(LowGamma_vect_F, :);
Global_spectrum_relpower = Mean_freq_chan_relpower(Global_vect_F, :);
OpenVibe_spectrum_relpower = Mean_freq_chan_relpower(OpenVibe_vect_F, :);

HighDelta_spectrum_percchange = Mean_freq_chan_percchange(HighDelta_vect_F, :);
Theta_spectrum_percchange = Mean_freq_chan_percchange(Theta_vect_F, :);
LowAlpha_spectrum_percchange = Mean_freq_chan_percchange(LowAlpha_vect_F, :);
HighAlpha_spectrum_percchange = Mean_freq_chan_percchange(HighAlpha_vect_F, :);
LowBeta_spectrum_percchange = Mean_freq_chan_percchange(LowBeta_vect_F, :);
HighBeta_spectrum_percchange = Mean_freq_chan_percchange(HighBeta_vect_F, :);
LowGamma_spectrum_percchange = Mean_freq_chan_percchange(LowGamma_vect_F, :);
Global_spectrum_percchange = Mean_freq_chan_percchange(Global_vect_F, :);
OpenVibe_spectrum_percchange = Mean_freq_chan_percchange(OpenVibe_vect_F, :);

```

```

% for all channels, extract the power values only in the parts of the spectrum that
are beta

```

```

HighDelta_mean_percchange = mean(HighDelta_spectrum_percchange,1);
Theta_mean_percchange     = mean(Theta_spectrum_percchange,1);
LowAlpha_mean_percchange  = mean(LowAlpha_spectrum_percchange,1);
HighAlpha_mean_percchange = mean(HighAlpha_spectrum_percchange,1);
LowBeta_mean_percchange   = mean(LowBeta_spectrum_percchange,1);
HighBeta_mean_percchange  = mean(HighBeta_spectrum_percchange,1);
LowGamma_mean_percchange  = mean(LowGamma_spectrum_percchange,1);
Global_mean_percchange    = mean(Global_spectrum_percchange,1);
OpenVibe_mean_percchange  = mean(OpenVibe_spectrum_percchange,1);

```

```

HighDelta_mean_relpower = mean(HighDelta_spectrum_relpower,1);
Theta_mean_relpower    = mean(Theta_spectrum_relpower,1);
LowAlpha_mean_relpower = mean(LowAlpha_spectrum_relpower,1);
HighAlpha_mean_relpower = mean(HighAlpha_spectrum_relpower,1);
LowBeta_mean_relpower  = mean(LowBeta_spectrum_relpower,1);
HighBeta_mean_relpower = mean(HighBeta_spectrum_relpower,1);
LowGamma_mean_relpower = mean(LowGamma_spectrum_relpower,1);
Global_mean_relpower   = mean(Global_spectrum_relpower,1);
OpenVibe_mean_relpower = mean(OpenVibe_spectrum_relpower,1);

```

```

% This might need adjustment: recursion of subj_data

```

```

subj_HighDelta_data_percchange = [pid, n, ses, block,
HighDelta_mean_percchange];
subj_Theta_data_percchange     = [pid, n, ses, block, Theta_mean_percchange];
subj_LowAlpha_data_percchange  = [pid, n, ses, block,
LowAlpha_mean_percchange];
subj_HighAlpha_data_percchange = [pid, n, ses, block,
HighAlpha_mean_percchange];
subj_LowBeta_data_percchange   = [pid, n, ses, block, LowBeta_mean_percchange];
subj_HighBeta_data_percchange  = [pid, n, ses, block,
HighBeta_mean_percchange];
subj_LowGamma_data_percchange  = [pid, n, ses, block,
LowGamma_mean_percchange];
subj_Global_data_percchange    = [pid, n, ses, block, Global_mean_percchange];
subj_OpenVibe_data_percchange  = [pid, n, ses, block,
OpenVibe_mean_percchange];

```

```

subj_HighDelta_data_relpower = [pid, n, ses, block, HighDelta_mean_relpower];
subj_Theta_data_relpower    = [pid, n, ses, block, Theta_mean_relpower];
subj_LowAlpha_data_relpower = [pid, n, ses, block, LowAlpha_mean_relpower];
subj_HighAlpha_data_relpower = [pid, n, ses, block, HighAlpha_mean_relpower];
subj_LowBeta_data_relpower  = [pid, n, ses, block, LowBeta_mean_relpower];
subj_HighBeta_data_relpower = [pid, n, ses, block, HighBeta_mean_relpower];
subj_LowGamma_data_relpower = [pid, n, ses, block, LowGamma_mean_relpower];
subj_Global_data_relpower   = [pid, n, ses, block, Global_mean_relpower];
subj_OpenVibe_data_relpower = [pid, n, ses, block, OpenVibe_mean_relpower];

```

```

%% Insert data into corresponding freq-block

```

```

HighDelta_data_percchange = [HighDelta_data_percchange ;
subj_HighDelta_data_percchange];
Theta_data_percchange     = [Theta_data_percchange ;
subj_Theta_data_percchange];
LowAlpha_data_percchange  = [LowAlpha_data_percchange ;
subj_LowAlpha_data_percchange];
HighAlpha_data_percchange = [HighAlpha_data_percchange ;
subj_HighAlpha_data_percchange];
LowBeta_data_percchange   = [LowBeta_data_percchange ;
subj_LowBeta_data_percchange];
HighBeta_data_percchange  = [HighBeta_data_percchange ;
subj_HighBeta_data_percchange];
LowGamma_data_percchange  = [LowGamma_data_percchange ;
subj_LowGamma_data_percchange];
Global_data_percchange    = [Global_data_percchange ;
subj_Global_data_percchange];
OpenVibe_data_percchange  = [OpenVibe_data_percchange ;
subj_OpenVibe_data_percchange];

HighDelta_data_relpower = [HighDelta_data_relpower ;
subj_HighDelta_data_relpower];
Theta_data_relpower     = [Theta_data_relpower ; subj_Theta_data_relpower];
LowAlpha_data_relpower  = [LowAlpha_data_relpower ;
subj_LowAlpha_data_relpower];
HighAlpha_data_relpower = [HighAlpha_data_relpower ;
subj_HighAlpha_data_relpower];

```

```

        LowBeta_data_relpower = [LowBeta_data_relpower ;
subj_LowBeta_data_relpower];
        HighBeta_data_relpower = [HighBeta_data_relpower ;
subj_HighBeta_data_relpower];
        LowGamma_data_relpower = [LowGamma_data_relpower ;
subj_LowGamma_data_relpower];
        Global_data_relpower = [Global_data_relpower ; subj_Global_data_relpower];
        OpenVibe_data_relpower = [OpenVibe_data_relpower ;
subj_OpenVibe_data_relpower];

        %% Also save matrices subject wise?
        save([save_dir 'Subj Mats/subj-' num2str(pid) '-ses-' num2str(ses) '-block-'
num2str(block) '-cond-' num2str(n) '-power_relpower.mat'], 'power_data_relpow');
        save([save_dir 'Subj Mats/subj-' num2str(pid) '-ses-' num2str(ses) '-block-'
num2str(block) '-cond-' num2str(n) '-power_percchange.mat'], 'power_data_percchange');

        %% Reset Loop
        % reselect original EEG.set to do other condition
        [ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 2, 'retrieve', 1, 'study', 0);
        toc
    end % End of one Condition, go to next

    catch
        disp(['skipping ' current_filename]);
        % analysis_diagnostics(size(analysis_diagnostics, 2)+1,1) = {'skipped '
current_filename ', n = ' num2str(n) 'Epo = ' num2str(size(EEG.data,3))}];
    end
    % delete files eegsets to not fill memory
    STUDY = []; CURRENTSTUDY = 0; ALLEEG = []; EEG = []; CURRENTSET = [];
    % close topoplots
    close all;
end % end of file loop
%% Save Power Range(rs) percchange
headings =
{'PID', 'UP1DN2', 'Ses', 'Block', 'Ch1', 'Ch2', 'Ch3', 'Ch4', 'Ch5', 'Ch6', 'Ch7', 'Ch8', 'Ch9', 'Ch10',
'Ch11', 'Ch12', 'Ch13', 'Ch14', 'Ch15', 'Ch16', ...

'Ch17', 'Ch18', 'Ch19', 'Ch20', 'Ch21', 'Ch22', 'Ch23', 'Ch24', 'Ch25', 'Ch26', 'Ch27', 'Ch28', 'Ch29',
'Ch30', 'Ch31', 'Ch32', ...

'Ch33', 'Ch34', 'Ch35', 'Ch36', 'Ch37', 'Ch38', 'Ch39', 'Ch40', 'Ch41', 'Ch42', 'Ch43', 'Ch44', 'Ch45',
'Ch46', 'Ch47', 'Ch48', ...

'Ch49', 'Ch50', 'Ch51', 'Ch52', 'Ch53', 'Ch54', 'Ch55', 'Ch56', 'Ch57', 'Ch58', 'Ch59', 'Ch60', 'Ch61',
'Ch62', 'Ch63', 'Ch64'}; %Headings for the summary Table that will be saved out

T_HighDelta = array2table(HighDelta_data_percchange, 'VariableNames', headings);
T_Theta = array2table(Theta_data_percchange, 'VariableNames', headings);
T_LowAlpha = array2table(LowAlpha_data_percchange, 'VariableNames', headings);
T_HighAlpha = array2table(HighAlpha_data_percchange, 'VariableNames', headings);
T_LowBeta = array2table(LowBeta_data_percchange, 'VariableNames', headings);
T_HighBeta = array2table(HighBeta_data_percchange, 'VariableNames', headings);
T_LowGamma = array2table(LowGamma_data_percchange, 'VariableNames', headings);
T_Global = array2table(Global_data_percchange, 'VariableNames', headings);
T_OpenVibe = array2table(OpenVibe_data_percchange, 'VariableNames', headings);

writetable(T_HighDelta, [save_dir 'percchange_HighDelta_' [num2str(HighDelta_min_freq) '_&_'
num2str(HighDelta_max_freq) '_Hz'] '_Power.csv']);
writetable(T_Theta, [save_dir 'percchange_Theta_' [num2str(Theta_min_freq) '_&_'
num2str(Theta_max_freq) '_Hz'] '_Power.csv']);
writetable(T_LowAlpha, [save_dir 'percchange_LowAlpha_' [num2str(LowAlpha_min_freq) '_&_'
num2str(LowAlpha_max_freq) '_Hz'] '_Power.csv']);
writetable(T_HighAlpha, [save_dir 'percchange_HighAlpha_' [num2str(HighAlpha_min_freq) '_&_'
num2str(HighAlpha_max_freq) '_Hz'] '_Power.csv']);
writetable(T_LowBeta, [save_dir 'percchange_LowBeta_' [num2str(LowBeta_min_freq) '_&_'
num2str(LowBeta_max_freq) '_Hz'] '_Power.csv']);
writetable(T_HighBeta, [save_dir 'percchange_HighBeta_' [num2str(HighBeta_min_freq) '_&_'
num2str(HighBeta_max_freq) '_Hz'] '_Power.csv']);
writetable(T_LowGamma, [save_dir 'percchange_LowGamma_' [num2str(LowGamma_min_freq) '_&_'
num2str(LowGamma_max_freq) '_Hz'] '_Power.csv']);
writetable(T_Global, [save_dir 'percchange_Global_' [num2str(Global_min_freq) '_&_'
num2str(Global_max_freq) '_Hz'] '_Power.csv']);
writetable(T_OpenVibe, [save_dir 'percchange_OpenVibe_' [num2str(OpenVibe_min_freq) '_&_'
num2str(OpenVibe_max_freq) '_Hz'] '_Power.csv']);

```

```

save([save_dir 'percchange_HighDelta_' [num2str(HighDelta_min_freq) '_&_'
num2str(HighDelta_max_freq) '_Hz'] '_Power.mat'], 'T_HighDelta');
save([save_dir 'percchange_Theta_' [num2str(Theta_min_freq) '_&_' num2str(Theta_max_freq)
'_Hz'] '_Power.mat'], 'T_Theta');
save([save_dir 'percchange_LowAlpha_' [num2str(LowAlpha_min_freq) '_&_'
num2str(LowAlpha_max_freq) '_Hz'] '_Power.mat'], 'T_LowAlpha');
save([save_dir 'percchange_HighAlpha_' [num2str(HighAlpha_min_freq) '_&_'
num2str(HighAlpha_max_freq) '_Hz'] '_Power.mat'], 'T_HighAlpha');
save([save_dir 'percchange_LowBeta_' [num2str(LowBeta_min_freq) '_&_'
num2str(LowBeta_max_freq) '_Hz'] '_Power.mat'], 'T_LowBeta');
save([save_dir 'percchange_HighBeta_' [num2str(HighBeta_min_freq) '_&_'
num2str(HighBeta_max_freq) '_Hz'] '_Power.mat'], 'T_HighBeta');
save([save_dir 'percchange_LowGamma_' [num2str(LowGamma_min_freq) '_&_'
num2str(LowGamma_max_freq) '_Hz'] '_Power.mat'], 'T_LowGamma');
save([save_dir 'percchange_Global_' [num2str(Global_min_freq) '_&_'
num2str(Global_max_freq) '_Hz'] '_Power.mat'], 'T_Global');
save([save_dir 'percchange_OpenVibe_' [num2str(OpenVibe_min_freq) '_&_'
num2str(OpenVibe_max_freq) '_Hz'] '_Power.mat'], 'T_OpenVibe');

%% Relpower
T_HighDelta = array2table(HighDelta_data_relpower, 'VariableNames', headings);
T_Theta = array2table(Theta_data_relpower, 'VariableNames', headings);
T_LowAlpha = array2table(LowAlpha_data_relpower, 'VariableNames', headings);
T_HighAlpha = array2table(HighAlpha_data_relpower, 'VariableNames', headings);
T_LowBeta = array2table(LowBeta_data_relpower, 'VariableNames', headings);
T_HighBeta = array2table(HighBeta_data_relpower, 'VariableNames', headings);
T_LowGamma = array2table(LowGamma_data_relpower, 'VariableNames', headings);
T_Global = array2table(Global_data_relpower, 'VariableNames', headings);
T_OpenVibe = array2table(OpenVibe_data_relpower, 'VariableNames', headings);

writetable(T_HighDelta, [save_dir 'relpower_HighDelta_' [num2str(HighDelta_min_freq) '_&_'
num2str(HighDelta_max_freq) '_Hz'] '_Power.csv']);
writetable(T_Theta, [save_dir 'relpower_Theta_' [num2str(Theta_min_freq) '_&_'
num2str(Theta_max_freq) '_Hz'] '_Power.csv']);
writetable(T_LowAlpha, [save_dir 'relpower_LowAlpha_' [num2str(LowAlpha_min_freq) '_&_'
num2str(LowAlpha_max_freq) '_Hz'] '_Power.csv']);
writetable(T_HighAlpha, [save_dir 'relpower_HighAlpha_' [num2str(HighAlpha_min_freq) '_&_'
num2str(HighAlpha_max_freq) '_Hz'] '_Power.csv']);
writetable(T_LowBeta, [save_dir 'relpower_LowBeta_' [num2str(LowBeta_min_freq) '_&_'
num2str(LowBeta_max_freq) '_Hz'] '_Power.csv']);
writetable(T_HighBeta, [save_dir 'relpower_HighBeta_' [num2str(HighBeta_min_freq) '_&_'
num2str(HighBeta_max_freq) '_Hz'] '_Power.csv']);
writetable(T_LowGamma, [save_dir 'relpower_LowGamma_' [num2str(LowGamma_min_freq) '_&_'
num2str(LowGamma_max_freq) '_Hz'] '_Power.csv']);
writetable(T_Global, [save_dir 'relpower_Global_' [num2str(Global_min_freq) '_&_'
num2str(Global_max_freq) '_Hz'] '_Power.csv']);
writetable(T_OpenVibe, [save_dir 'relpower_OpenVibe_' [num2str(OpenVibe_min_freq) '_&_'
num2str(OpenVibe_max_freq) '_Hz'] '_Power.csv']);

save([save_dir 'relpower_HighDelta_' [num2str(HighDelta_min_freq) '_&_'
num2str(HighDelta_max_freq) '_Hz'] '_Power.mat'], 'T_HighDelta');
save([save_dir 'relpower_Theta_' [num2str(Theta_min_freq) '_&_' num2str(Theta_max_freq)
'_Hz'] '_Power.mat'], 'T_Theta');
save([save_dir 'relpower_LowAlpha_' [num2str(LowAlpha_min_freq) '_&_'
num2str(LowAlpha_max_freq) '_Hz'] '_Power.mat'], 'T_LowAlpha');
save([save_dir 'relpower_HighAlpha_' [num2str(HighAlpha_min_freq) '_&_'
num2str(HighAlpha_max_freq) '_Hz'] '_Power.mat'], 'T_HighAlpha');
save([save_dir 'relpower_LowBeta_' [num2str(LowBeta_min_freq) '_&_'
num2str(LowBeta_max_freq) '_Hz'] '_Power.mat'], 'T_LowBeta');
save([save_dir 'relpower_HighBeta_' [num2str(HighBeta_min_freq) '_&_'
num2str(HighBeta_max_freq) '_Hz'] '_Power.mat'], 'T_HighBeta');
save([save_dir 'relpower_LowGamma_' [num2str(LowGamma_min_freq) '_&_'
num2str(LowGamma_max_freq) '_Hz'] '_Power.mat'], 'T_LowGamma');
save([save_dir 'relpower_Global_' [num2str(Global_min_freq) '_&_' num2str(Global_max_freq)
'_Hz'] '_Power.mat'], 'T_Global');
save([save_dir 'relpower_OpenVibe_' [num2str(OpenVibe_min_freq) '_&_'
num2str(OpenVibe_max_freq) '_Hz'] '_Power.mat'], 'T_OpenVibe');

```

10.4.4 Automatic Preprocessing of EEG Data using ASR and CSD (MATLAB)

This script automatically preprocesses EEG Data using ASR. It also Applies a CSD.

```
clear all

%% Setting up the structure
% Get Shared Folder
shared_dir = ''

save_dir = ''

root_folder = '';

% Prepare CSD
pre_root_path = cd('');
addpath(pre_root_path);
noEXTlocation_file = '';
noEXTlocation_file_csd = noEXTlocation_file;
noEXTlocation_file_csd(end-2:end) = 'csd';
ConvertLocations(noEXTlocation_file);
chanlocs = readlocs(noEXTlocation_file);
chanlabels = {chanlocs.labels};
chanlabels = cellfun(@upper, chanlabels, 'UniformOutput', false);
labels=chanlabels';
M = ExtractMontage(noEXTlocation_file_csd, labels);
% MapMontage(M);
[G,H] = GetGH(M);

work_dir = '';
cd(work_dir);

% do you want to run AMICA? if yes load eeglab without spaces
runAMICA = 0;
if runAMICA
    addpath('')
else
    addpath(''); %eeglab
end

subj_list= {''};

condition_list= {'UP', 'DN'};

% chanloc_file= [root_folder filesep 'BWB_chanlocs_eeglab.mat'];
% load(chanloc_file);

%% Preprocessing Diagnostics
% this one will catch the diagnostic data from the whole script
preprocessing_diagnostics = {};

% This one will catch diagnostic data from individual subjects
subj_diagnostics = {};

% This one will catch diagnostic data from individual subjects
eog_diagnostics = {};

% this variable will count how many files should be openend
files_check = 0;

%% Reading and doing Preprocessing
%open EEGlab
[ALLEEG EEG CURRENTSET ALLCOM] = eeglab;

% for j=1:length(subj_list)
j = 1;
% Find participant folder
current_filepath = ['./', 'EEGBCI EDF', '/']

cd(current_filepath) %All the sessions and blocks are in here. Treat every file the same
for now- can organise into ses and blocks afterwards
```

```

contents = dir('*online*.edf');
dirFlags = ~[contents.isdir] & ~strcmp({contents.name}, '.') & ~strcmp({contents.name}, '..')
& ~strcmp({contents.name}, '.DS_Store');
files=contents(dirFlags); %list only the files that are real

subj_data=[]; %empty vector for this subj

% update file_sum
files_check = length(files);

debug_start = 1;
debug_end = length(files);
part = [num2str(debug_start), '-', num2str(debug_end)];
length(files);
%% Cycle through participant files
for k = debug_start:debug_end %for every file in the folder, apply the analysis and save
results
    % ASR has not run yet
    ASR_ran = false;
    % ICA has not run yet;
    ICA_ran = false;
    % File is epoched and saved;
    EPO_ran = false;
    try
        current_filename=files(k).name;
        % Extract the session and block numbers
        pid = regexp(current_filename, 'copy-p(\d+)', 'tokens');
        pid = str2double(pid{1}{1});

        ses = regexp(current_filename, 'sess-?(\d+)', 'tokens');
        ses = str2double(ses{1}{1});

        block = regexp(current_filename, 'block(\d+)|b(\d+)\.', 'tokens');
        block = str2double(block{1}{1});

        %determine new filename
        new_filename = ['sub-' num2str(pid) '-taskOnline-sess' num2str(ses) '-block'
num2str(block) '-eeg.set'];

        %Below is to import the file to be processed
        %load file
        EEG = pop_biosig([current_filename]);
        [ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG,
1, 'setname', 'loaded_gdf', 'gui', 'off');
        EEG = eeg_checkset( EEG );

        EEG = pop_select( EEG, 'time', [10 316] );
        [ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 2, 'gui', 'off');

        %% Try AvRef ASR AvRef with Interpol
        try

            % Go through all events and give the target markers 1s more
            % to make them more detectable in epochs
            for n = 1:size(EEG.event,2)

                % Check name of event for right hand cue
                if strcmp(EEG.event(n).type, 'OVTK_GDF_Right')
                    % add latency so that event is pushed by 1s
                    % (= EEG sampling rate)
                    EEG.event(n).latency = EEG.event(n).latency + EEG.srate;

                    % Check for left hand cue
                elseif strcmp(EEG.event(n).type, 'OVTK_GDF_Left')
                    % add latency so that event is pushed by 1s
                    EEG.event(n).latency = EEG.event(n).latency + EEG.srate;

                end
            end

            % Downsampling for manageable speed
            EEG = pop_resample( EEG, 250);
            [ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 3, 'gui', 'off');

            % notch filter
            EEG = pop_eegfiltnew(EEG,
'locutoff', 48, 'hicutoff', 52, 'revfilt', 1, 'plotfreqz', 0);
            [ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 4, 'gui', 'off');

```

```

% 1-40 PB filter
EEG = pop_eegfiltnew(EEG, 'locutoff',1,'hicutoff',45,'plotfreqz',0);
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 5,'gui','off');

% relabel 15, 28 and 52, because they come with additional
% labels that eeglab cannot interpret
EEG=pop_chanedit(EEG,
'changefield',{15,'labels','T7'},'changefield',{28,'labels','Iz'},'changefield',{52,'labels','T8'});
[ALLEEG EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);

% Set Channel locations
EEG=pop_chanedit(EEG, 'lookup',' ');
[ALLEEG EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);

try
%% Let's try to incorporate the EOG-corrected EEG
% load in b-matrix
eog_path = ['./nf' num2str(pid) '/EEG BCI/EOG/'];
eog_path = dir([eog_path 'b-Matrix*' num2str(ses) '*.cfg']);
eog_path = ['./nf' num2str(pid) '/EEG BCI/EOG/', eog_path.name];

fid = fopen(eog_path, 'r');
eog_mat = textscan(fid, '%s %s', 'delimiter', '=');
fclose(fid);

% transform file info into actual eog_mat
eog_mat = eog_mat{1};
eog_firstrow = eog_mat(1);
eog_mat(1) = [];
eog_mat = str2double(eog_mat);
eog_mat = reshape(eog_mat, 4, 64);
eog_mat2 = reshape(eog_mat, 64, 4);

% create EEG data (take just EEG data from EEG electrodes)
EEG_uncorr = EEG.data(1:64,:);

% create EOG data (take EOG data from facial EOG electrodes)
EOG_chan = EEG.data(65:68,:);

% create data openvibe has seen
EEG_corrected_1 = EEG_uncorr - eog_mat*EOG_chan;
EEG_corrected_2 = EEG_uncorr - eog_mat2*EOG_chan;

% give back transformed data
EEG.data(1:64,:) = EEG_corrected_1;
catch
disp(['skipping EOG for' current_filename]);
% Diagnostics
eog_diagnostics(length(eog_diagnostics)+1,:) = {'skipped EOG for '
current_filename}};
continue
end
%% Continue with ASR
% Avref
EEG = pop_reref( EEG, []);
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 6,'gui','off');
EEG = eeg_checkset( EEG );

% Exclude extra electrodes
EEG = pop_select( EEG, 'nochannel',{'EX 1','EX 2','EX 3','EX 4','EX 5','EX 6','EX 7','EX 8'});
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 7,'gui','off');

%preASR chan number
preASR_channr = EEG.nbchan;
% need to save urchanlocs, as it doesn't save itself.
originalEEG = EEG;

% ASR
EEG = pop_clean_rawdata(EEG,
'FlatlineCriterion',5,'ChannelCriterion',0.8,'LineNoiseCriterion',4,'Highpass','off','Burst
Criterion',20,'WindowCriterion','off','BurstRejection','off','Distance','Euclidian');
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 8,'gui','off');
EEG = eeg_checkset( EEG );

```



```

% do ASR Diagnostics
ASR_ran = true;
% How many cue markers are left after ASR
preASR_events = struct2table(EEG.urevent);

% find trials
% Search for 'OVTK_GDF_Right' and 'OVTK_GDF_Left'
idxRight = cellfun(@(x) ~isempty(strfind(x, 'OVTK_GDF_Right')),
preASR_events.type);
idxLeft = cellfun(@(x) ~isempty(strfind(x, 'OVTK_GDF_Left')),
preASR_events.type);
preASR_trials = sum(idxLeft) + sum(idxRight);

postASR_events = struct2table(EEG.event);
idxRight = cellfun(@(x) ~isempty(strfind(x, 'OVTK_GDF_Right')),
postASR_events.type);
idxLeft = cellfun(@(x) ~isempty(strfind(x, 'OVTK_GDF_Left')),
postASR_events.type);
postASR_trials = sum(idxLeft) + sum(idxRight);

% How many missing channels
postASR_channr = EEG.nbchan;

% AvRef & interpolating missing channels
% Missing Electrode Names
int_elec = setdiff({originalEEG.chanlocs.labels}, {EEG.chanlocs.labels});
EEG = pop_interp(EEG, originalEEG.chanlocs, 'spherical');
% Avref
EEG = pop_reref( EEG, []);
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 9, 'gui', 'off');
EEG = eeg_checkset( EEG );

% Data ASR kept – stopped working with updated code
eeg_kept = 1; % mean(EEG.etc.clean_sample_mask);
eeg_before = 1; % originalEEG.data(:,EEG.etc.clean_sample_mask);
eeg_PVAF = {};
% eeg_PVAF = 100-100*mean(var(eeg_before-EEG.data))/mean(var(eeg_before));

% enter Diagnostics catch if anything doesn't work.
catch % ASR Catch
disp(['skipping ASR for' current_filename]);
% Diagnostics
preprocessing_diagnostics(length(preprocessing_diagnostics)+1,:) = {'skipped
ASR for ' current_filename}};

end % end of try ASR

%% Try Automatic ICA & Component labeling & rejection
try
% PICARD is a faster ICA than the RUNICA, but the same apparently
% EEG = pop_runica(EEG, 'pca', 'icatype', 'picard', 'maxiter',500);

if runAMICA
previous_path = pwd;
cd('/Applications/MATLAB/Matlab_Toolboxes/')

numprocs = 1; % # of nodes
max_threads = 1; % # of threads
num_models = 1; % # of models of mixture ICA
max_iter = 1000; % max number of learning steps
% run amica
outdir = [ '/Applications/MATLAB/Matlab_Toolboxes/' filesep 'amicaouttmp'
filesep ];
[weights,sphere,mods] = runamica15(EEG.data, 'num_models',num_models,
'outdir',outdir, ...
'numprocs', numprocs, 'max_threads', max_threads, 'max_iter',max_iter);

EEG =
pop_loadmodout(EEG, '/Applications/MATLAB/Matlab_Toolboxes/amicaouttmp/')
[ALLEEG EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);

cd(previous_path)
else
EEG = pop_runica(EEG, 'icatype', 'picard', 'maxiter',500, 'pca',
postASR_channr);
EEG = eeg_checkset( EEG );
end

```



```

% Automatically label ICA components
EEG = pop_iclabel(EEG, 'default');
EEG = eeg_checkset( EEG );

% Flag suspicious ICA Components for rejection (from
% certainty 90% up)
EEG = pop_icflag(EEG, [NaN NaN;0.9 1;0.9 1;0.9 1;0.9 1;0.9 1;0.9 1]);
EEG = eeg_checkset( EEG );

% Diagnostics: Which Components to reject, before
% rejection
% How many components were rejected:
rej_nr = sum(EEG.reject.gcompreject);
% find out what label and index
rej_index = find(EEG.reject.gcompreject);
[rej_val_max, rej_val_max_ind] =
max(EEG.etc.ic_classification.ICLabel.classifications(rej_index,:), [], 2);
rej_labels = EEG.etc.ic_classification.ICLabel.classes(1, rej_val_max_ind);

% join rej_index and rej_labels to one cell so we can
% add it to diagnostics
rej_comp = {rej_index rej_labels'}

% ICA ran
ICA_ran = true;

% Remove flagged ICA components
EEG = pop_subcomp( EEG, [], 0);
EEG = eeg_checkset( EEG );

catch % Catch of automatic ICA

    disp(['skipping ICA for' current_filename]);
    % Diagnostics
    preprocessing_diagnostics(length(preprocessing_diagnostics)+1,:) = {'skipped
ICA for ' current_filename}];

end % end of Automatic ICA

%% Epoching & Saving File:
try
    % Epoch at the Cue marker + 2s rest period (has been pushed
    % to 1s. For StabDist analysis we need 2 s rest. so we go
    % to -3s. but we also need 2s of signal (but have 3.75). If we ignore the
marker
    % space then we need to go from marker point + 0.25 (end of
    % marker time) + 2 seconds, so 2.25. but then we only take
    % about 2s just after marker time and ignore the rest of 1.75s
    % of signal time. Let's take all of it so 0.25s + 3.75 = 4s
    % EEG = pop_epoch( EEG, { 'Start of Trial, Trigger at t=0s' }, [4.25
8], 'newname', 'loaded_gdf pruned with ICA epochs', 'epochinfo', 'yes');
    EEG = pop_epoch( EEG, { 'OVTK_GDF_Left' 'OVTK_GDF_Right' }, [-3 4],
'newname', 'loaded_gdf pruned with ICA epochs', 'epochinfo', 'yes');
    [ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 10, 'gui', 'off');
    EEG = eeg_checkset( EEG );

    % This discards uncontinuous trials. Epoching around
    % 1.25-5s of trial might leave us with more trials!

    post_epoch_trials = size(EEG.epoch, 2);

    % % Saving the file as EDF
    % pop_wroteeeg(EEG, [new_filepath filesep new_filename],
'TYPE', 'EDF');
    % save as set:
    EEG = pop_saveset( EEG, 'filename', new_filename, 'filepath', save_dir);
    [ALLEEG EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);
    EEG = eeg_checkset( EEG );

    EPO_ran = true;
catch % Catch of Epoching and saving

    disp(['skipping EPO for' current_filename]);
    % Diagnostics
    preprocessing_diagnostics(length(preprocessing_diagnostics)+1,:) = {'skipped
EPO for ' current_filename}];

end % End of Epoching & saving

```

```

%% Diagnostics if it ran
if (ASR_ran && ICA_ran && EPO_ran)
    % compile Diagnostics
    subj_diagnostics(height(subj_diagnostics)+1,:) = {[num2str(pid), '_'],
current_filename], new_filename, preASR_trials, postASR_trials, post_epoch_trials,
preASR_channr, postASR_channr, num2str(rej_index'), strjoin(int_elec, ', '), eeg_kept*100,
eeg_PVAF];
    preprocessing_diagnostics(length(preprocessing_diagnostics)+1,:) = {'ran '
current_filename}];
    % rej_table = table(rej_comp, 'VariableNames', {'Index' 'Reason for
Rejection'})
end

%% CSD (before epoching)
% Delete external channels from current dataset
data = repmat(NaN,size(EEG.data)); % use single data precision
% loop through all epochs
myEEG = single(EEG.data(:,:,)); %if looping through epochs add EEG.data(:,:,ne)
% reduce data precision to reduce memory demand
MyResults = CSD(myEEG,G,H); % compute CSD for <channels-by-samples> 2-D
epoch
data(:,:,) = MyResults; % assign data output. if looping through epochs
add data(:,:,ne)

EEG.data=data;
EEG = eeg_checkset( EEG );
% Delete all sets out of Memory
STUDY = []; CURRENTSTUDY = 0; ALLEEG = []; EEG=[]; CURRENTSET=[];
end % end of try to do this file

end % end of file loop

% check if the amount of entries in the diagnostics table match the
% subj sum. If not, add the difference into diagnostics file:
% number of files saved
content_check = length(dir([save_dir filesep '*.set']));
if ~(files_check == content_check)
    preprocessing_diagnostics(length(preprocessing_diagnostics)+1,:) = {'Participant '
num2str(pid), ' has: ', num2str(files_check), ' files to process, but only: ',
num2str(content_check) ' files processed.' }];
end

close all %

% write global summary
writecell(preprocessing_diagnostics, [save_dir filesep 'Preprocessing_Diagnostics_part'
part '.xlsx'])

% write specific data report per subject
subj_table = cell2table(subj_diagnostics, 'VariableNames', {'Old Filename' 'New Filename'
'Pre ASR Epochs' 'Post ASR Epochs' 'Post Epoching Trials' 'Pre ASR Channels' 'Post ASR
Channels' 'Index of Rejected Components' 'Name of the rejected Electrodes' '% of EEG kept
by ASR' 'PVAF of ASR'});
writetable(subj_table, [save_dir filesep 'Preprocessing_Details_part' part '.xlsx']);

% write eeg tabe
writecell(eeg_diagnostics, [save_dir filesep 'EOG_Diagnostics_part' part '.xlsx']);

% Write a report on the Specs the Preprocessing ran with:
% writecell(Preprocessing_Parameters', [save_dir filesep 'Preprocessing_Parameters_'
datestr(datetime) '.txt'])

%% Notes:
% Exclude electrodes. Check why so much gets rejected.

```

10.4.5 Stability and Distinctiveness (MATLAB)

This Script reads in Epoched EEG data and computes stability and distinctiveness.

```
% this is to bring data into shape for new metrics
%% clear all
clear all;
clc;

%% Setup
addpath('');
addpath('');

% add EEGLab
addpath(''); %eeglab
% Activate riemann toolbox from here:
https://github.com/alexandrebarachant/covariancetoolbox.git
% run('')
run('');

data_dir = '';
work_dir = '';
save_dir = '';

% load in data:
subj_folders = dir([data_dir filesep 'sub*.set']);

% set working directory
cd(work_dir);

%% Start EEGLAB
% open EEGLab
[ALLEEG EEG CURRENTSET ALLCOM] = eeglab;

%% Timings
% It might actually complain that it they all need to be the same length
% set sampling rate:
EEG_srate = 250;
% set resting time (before trial begins)
rest_time = 2;
% choose the smallest data spot:
rest_index = rest_time*EEG_srate;

% conditions
EEG_srate = 250;
EEG_size = 7 * EEG_srate;
% so we get whole data period
EEG_snip = floor(0.375 * EEG_srate);

Baseline_idx = [1 2*EEG_srate];
Condition_idx = [((EEG_size - EEG_snip) - 3 * EEG_srate) ...
    (EEG_size - EEG_snip)];

% init results mat
current_results = [];
error_msgs = {};

%% Get Subj Numbers
% get subj nrs
file_names = {subj_folders.name}';
% Define the regular expression pattern to match the numbers in the file names
pattern = 'sub-(\d+)-';

% Initialize an empty array to store the extracted numbers
nf_numbers = [];

% Loop through each file name in the cell array
for i = 1:length(file_names)
    % Use the regular expression to extract the number from the file name
    match = regexp(file_names{i}, pattern, 'tokens');

    % If a match is found, append the extracted number to the xx_numbers array
    if ~isempty(match)
```

```

        number = str2double(match{1}{1});
        nf_numbers(end+1) = number;
    end
end
nf_numbers = unique(nf_numbers);

%% Start Processing
% for all participants
for subj_i = 1:length(nf_numbers)

    % for all the files
    current_files = dir([data_dir '*sub-' num2str(nf_numbers(subj_i)) '-*.set']);
    for subj_files = 1:size(current_files, 1)

        current_filename = current_files(subj_files).name;
        current_filepath = current_files(subj_files).folder;

        [ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 0, 'gui', 'off'); %
        EEG = pop_loadset('filename', current_filename, 'filepath',
current_filepath); %load the data .set file
        EEG = eeg_checkset( EEG );

        % skip if empty
        if isempty(EEG.epoch)
            errormsg = [current_filename ' has no epochs'];
            error_msgs(length(error_msgs) + 1) = {errormsg};
            continue
        end

        % transform trials into sensible datatype, thanks matlab for not
        % offering an easy way to do this.
        % EEGDataRest.y: a [1 * Nt] vector containing the class labels.
        % This is already correct format.
        trials = {EEG.epoch.eventtype}';
        for trials_n = 1:size(trials, 1)
            trial_name(trials_n) = {trials{trials_n}{2}};
        end
        clear('trials', 'trials_n');

        trials = {EEG.epoch.eventedftype}';
        for trials_n = 1:size(trials, 1)
            trial_type(trials_n) = trials{trials_n}{2};
        end
        clear('trials', 'trials_n');

        % Now let's create the correct data frame
        % EEGDataRest.x: the EEG signals as a [Ns * Nc * Nt] Matrix where
        %     Ns: number of EEG samples per trial
        %     Nc: number of channels (EEG electrodes)
        %     Nt: number of trials
        % thich should change samples to be rows, and channels to be
        % columns, while leaving trial numbers intact.
        EEG_data = permute(EEG.data, [2, 1, 3]);

        % Select the appropriate datasections
        % first 2 seconds are resting time
        EEG_data_rest = EEG_data(Baseline_idx(1):Baseline_idx(2), :, :);

        % we have 3.75s of trial signal time. take the middle of that so
        % thats 3.75-0.875 for the end of the 2s period
        % find end of trial and go back 2 s
        end_trial_start = Condition_idx(1);
        end_trial_end = Condition_idx(2);

        % check and report if end of trial is within borders
        if end_trial_end > size(EEG_data, 1)
            errormsg = [current_filename, ' does not have enough sample: '
num2str(end_trial_end)];
            error_msgs(length(error_msgs)+1) = {errormsg};
            disp(errormsg)
        end

        EEG_data_trial = EEG_data(end_trial_start:end_trial_end, :, :);

        % Compute StabDist
        EEGDataRest = struct;
        EEGDataRest.x = EEG_data_rest;

```

```

EEGDataRest.y = trial_type;
EEGDataRest.s = EEG_srate;

EEGDataTask = struct;
EEGDataTask.x = EEG_data_trial;
EEGDataTask.y = trial_type;
EEGDataTask.s = EEG_srate;

%restDis: a vector with Ncl elements, Ncl being the number of classes in the data.
Each element is the distance between the class mean and the rest mean
%classDis: a value indicating the distinctiveness between all classes (whether for
2 or more classes).
%restStab: the stability (inverse of the variance) of the covariance matrices for
the rest class (a single value)
%classStab: a vector with each element being the the stability (inverse of the
variance) covariance matrices for the corresponding class
%classLabels: the classes labels of the Ncl classes (as indexed in restDis,
classDis and classStab)
try
    [restDis classDis restStab classStab classLabels] = ...
        computeMultiClassDistinctiveness(EEGDataRest, EEGDataTask);
catch
    errormsg = [current_filename ' Did not work. Filled in NA.'];
    disp(errormsg);
    error_msgs(length(error_msgs)+1) = {errormsg};
    error_msgs(length(error_msgs)+1) = {[unique(trial_type)]};
    continue
end
% put into results table:
Task_Distance(subj_files) = classDis;
Rest_Stability(subj_files) = restStab;

% here make sure that we always order values the same way:
% 770 is up and right and 769 is left and down.
% in this new version the numbers are not clearly labelled. find
% class by character.
trial_up = find(~cellfun(@isempty, strfind(trial_name, 'Right')));
trial_up = trial_type(trial_up(1));
trial_dn = find(~cellfun(@isempty, strfind(trial_name, 'Left')));
trial_dn = trial_type(trial_dn(1));
task_UP_ind = find(classLabels == trial_up);
task_DN_ind = find(classLabels == trial_dn);

%collect other info:
rest_taskUP_Dist(subj_files) = restDis(task_UP_ind);
rest_taskDN_Dist(subj_files) = restDis(task_DN_ind);

taskUP_Stab(subj_files) = classStab(task_UP_ind);
taskDN_Stab(subj_files) = classStab(task_DN_ind);

% get block and sess string
name_profile = strsplit(current_filename, '-');
ses(subj_files) = str2num(name_profile{4}(5));
block(subj_files) = str2num(name_profile{5}(6));
PID(subj_files) = str2num(name_profile{2});

% delete files eegsets to not fill memory
STUDY = []; CURRENTSTUDY = 0; ALLEEG = []; EEG=[]; CURRENTSET=[];

% clear variables for clean run
n_trials(subj_files) = length(trial_type);
occur_up(subj_files) = sum(trial_type == trial_up);
occur_down(subj_files) = sum(trial_type == trial_dn);
clear('trial_type');

end % End of reading all files

new_results_mat = [PID', ses', block', n_trials', occur_up', occur_down',
Task_Distance', rest_taskUP_Dist', ...
rest_taskDN_Dist', Rest_Stability', taskUP_Stab', taskDN_Stab'];

clear('PID', 'ses', 'block', 'n_trials', 'occur_up', 'occur_down', 'Task_Distance',
'rest_taskUP_Dist', ...
'rest_taskDN_Dist', 'Rest_Stability', 'taskUP_Stab', 'taskDN_Stab')

current_results = [current_results; new_results_mat];

%% Save Results

```

```

end % End of cycling through subject folders

%% Bring data into shape
table_headers = {'PID', 'Session', 'Block', '# of Trials', '# of Up', '# of Down',
'Distance UP:DN', ...
'Distance Rest:UP', 'Distance Rest:DN', 'Stability of Rest', ...
'Stability of UP', 'Stability of DN'};

result_table = array2table(current_results, 'VariableNames', table_headers);
writetable(result_table, [save_dir filesep 'Stability_Distance_Results.xlsx']);

diagnostics_table = cell2table(error_msgs);
writetable(diagnostics_table, [save_dir filesep 'Diagnostics.xlsx']);

```

10.4.6 QS Data (R)

This Script reads in data from multiple QS and combines them

```
# Script to read in KVIQ Scores and BCI CA
# This script reads in a few different files (namely, CA, KVIQ and KSS files),
# and merges together into a dataframe that can be used with MEPs or other data

# Init -----
#clear all
rm(list = ls())

library(readxl)
library(readr)

# Go into Folder
work_dir = ""
setwd(work_dir)
source(file = "../../../Code/R Code/R_useful_funcs_share.R")

# CA -----
# read in table
CA = read_excel("../2 Stage BCI Classification accuracy pc.xlsx", sheet = "Online")

# Only take PID, Sess, Block, CA, Ind Scores
df.ca = data.frame(cbind(CA$Participant, CA$Session, CA$Block, CA$`Accuracy [%]`,
                        CA$`L success (White)`, CA$`L failure (Green)`,
                        CA$`R failure (Green)`, CA$`R success (White)`)
colnames(df.ca) = c("PID", "Sess", "Block", "CA", "Success Cross", "Failure Cross",
                    "Failure Hand", "Success Hand")
df.ca$`Failure Bias` = as.numeric(df.ca$`Success Hand`) - as.numeric(df.ca$`Success Cross`)
df.ca$Unbalanced = ifelse(abs(df.ca$`Failure Bias`) == 15, 1, 0)
df.ca$UID = interaction(df.ca$PID, df.ca$Sess, df.ca$Block)

# KVIQ -----
KVIQ = read_csv(file = "../../../KVIQ Results and Randomisation/Results/KVIQ_Scores_FINAL.csv")

#Make Handedness into a score
EdinHandedness =
as.data.frame(KVIQ[,c("PleaseIndicateYourPreferencesInTheUseOfHandsInTheFollowingActiv",
"PleaseIndicateYourPreferencesInTheUseOfHandsInTheFollowingAct_1",
"PleaseIndicateYourPreferencesInTheUseOfHandsInTheFollowingAct_2",
"PleaseIndicateYourPreferencesInTheUseOfHandsInTheFollowingAct_3"))]
EdinHandedness$PleaseIndicateYourPreferencesInTheUseOfHandsInTheFollowingActiv =
factor(EdinHandedness$PleaseIndicateYourPreferencesInTheUseOfHandsInTheFollowingActiv,
       labels = c(1, 2, 3, 4, 5), levels = c("Always left", "Usually left",
       "Both equally", "Usually right", "Always right"))
EdinHandedness$PleaseIndicateYourPreferencesInTheUseOfHandsInTheFollowingAct_1 =
factor(EdinHandedness$PleaseIndicateYourPreferencesInTheUseOfHandsInTheFollowingAct_1,
       labels = c(1, 2, 3, 4, 5), levels = c("Always left", "Usually left",
       "Both equally", "Usually right", "Always right"))
EdinHandedness$PleaseIndicateYourPreferencesInTheUseOfHandsInTheFollowingAct_2 =
factor(EdinHandedness$PleaseIndicateYourPreferencesInTheUseOfHandsInTheFollowingAct_2,
       labels = c(1, 2, 3, 4, 5), levels = c("Always left", "Usually left",
       "Both equally", "Usually right", "Always right"))
EdinHandedness$PleaseIndicateYourPreferencesInTheUseOfHandsInTheFollowingAct_3 =
factor(EdinHandedness$PleaseIndicateYourPreferencesInTheUseOfHandsInTheFollowingAct_3,
       labels = c(1, 2, 3, 4, 5), levels = c("Always left", "Usually left",
       "Both equally", "Usually right", "Always right"))
HandednessScore = rowSums(cbind(as.numeric(EdinHandedness[,1]),
                                as.numeric(EdinHandedness[,2]),
                                as.numeric(EdinHandedness[,3]),
                                as.numeric(EdinHandedness[,4])))/4
```

```

# take sum scores, ID, sex, age, handedness items
df.kviq = data.frame(cbind(KVIQ$ParticipantID, KVIQ$WhichSexWhereYouAssignedAtBirth_,
KVIQ$HowOldAreYou_,
                        KVIQ$WhatIsYourDominantHand_, HandednessScore, KVIQ$SumScoreInt,
KVIQ$SumScoreImg, KVIQ$TotalScore))

colnames(df.kviq) = c("PID", "Sex", "Age", "Dominant Hand", "Dominant Hand Score",
                      "KVIQ Intensity", "KVIQ Imagery", "KVIQ Total")

# make PID numeric so it's works with other functions
df.kviq$PID = parse_number(df.kviq$PID)

## Investigations - Demographics -----
df.kviq$PID
#nr 10 is missing
table(df.kviq$Sex)
mean(as.numeric(df.kviq$`Dominant Hand Score`), na.rm = TRUE)

# KSS -----
KSS = read_excel("../KSS and Order.xlsx", sheet = "KSS Score")

df.KSS = KSS[, -5]
# homogenise values
df.KSS$KSS[which(df.KSS$KSS == "8.5/9")] = 9

# make numeric
df.KSS$KSS = as.numeric(df.KSS$KSS)

# Exclude participant 8 dropped out after a session
df.KSS = df.KSS[-c(which(df.KSS$Participant == 8)),]

# rename participant to pid
colnames(df.KSS) = c("PID", "Sess", "Block", "KSS")

# create UID for merging
df.KSS$UID = interaction(df.KSS$PID, df.KSS$Sess, df.KSS$Block)

# Synthesis -----

# Merge the dfs
df.kss.ca = merge_unique_cols(df.KSS, df.ca, "UID")
df.merged = merge_unique_cols(df.kss.ca, df.kviq, "PID")

# export
save(df.merged, file = "../Processed Data/QS Data/KVIQ-KSS-CA.Rdata")
write.csv(df.merged, file = "../Processed Data/QS Data/KVIQ-KSS-CA.csv")

```


10.4.7 TILDA Plot (R)

This script generates some of the plots used in the MRI paper (Chapter 7). A large part of this code has been commented out to update plots.

```
# clean all
rm(list=ls())

# # Run the logReg on all AFD and FA brain regions.
library(plyr)
library(caret)
library(psc1)
library(car)
library(ggplot2)
library(ggsci)
library(ggforce)
library(see)
library(extrafont)
library(pwr)
library(ggpubr)

# set wp
setwd("")

# choose to make raw or zScore plots
raw_or_zScore = "zScore"

if (raw_or_zScore == "zScore"){
  # # read in dataset:
  load("../..../")
  # rename control df
  control = df
  load("../")
}else{
  # read in raw data
  control = load("")
  # rename control df
  control = df
  load("")
}

source("")
save_path = paste("", raw_or_zScore, "/",
  raw_or_zScore, '_PA', sep = "")
save_control_path = paste("", raw_or_zScore, "/",
  "/control/", raw_or_zScore, '_PA', sep = "")
save_prediction_path = paste("", raw_or_zScore, "/",
  "/prediction/", raw_or_zScore, '_PA', sep = "")
save_int_path = paste("", raw_or_zScore, "/",
  "/int/", raw_or_zScore, '_PA', sep = "")

## Functions for Plots -----
# trim to exclude outliers (sd)
ID.outliers = function(data.vector, multiplier){
  # outliers are whatever is multiplier times bigger than mean
  data.mean = mean(data.vector, na.rm = TRUE)
  data.sd = sd(data.vector, na.rm = TRUE)
  upper.thresh = data.mean + data.sd*multiplier
  lower.thresh = data.mean - data.sd*multiplier
  result = data.vector > upper.thresh | data.vector < lower.thresh
  return(result)
}
```

```

df$sex_W3 = factor(df$sex_W3, levels = c(1, 2), labels = c("Male", "Female"))
df$FRexercise3_W3 = factor(df$FRexercise3_W3, levels = c("0", "1", "2"), labels = c("Low
Activity", "Moderate Activity", "High Activity"))

# Create Plot DF -----

# exclude above 3
ggdf.FOD = df
ggdf.FOD$FOD.outliers = ID.outliers(ggdf.FOD$rIFG_31.FOD, 3)
ggdf.FOD = data.frame(df$fallers_w3, df$rIFG_31.FOD, df$age_W3, df$BP_W3,
  ggdf.FOD$FOD.outliers, df$FRexercise3_W3)
colnames(ggdf.FOD) = c("Fallers", "rIFG.31.FOD", "Age", "BloodPressure", "Outliers",
"PhysicalActivity")
ggdf.FOD = ggdf.FOD[!is.na(ggdf.FOD$rIFG.31.FOD),]
ggdf.FOD = ggdf.FOD[!ggdf.FOD$Outliers, ]
ggdf.FOD$Fallers = factor(ggdf.FOD$Fallers, levels = c(0, 1), labels = c("Not Faller",
"Faller"))
ggdf.FOD = ggdf.FOD[!is.na(ggdf.FOD$PhysicalActivity), ]
ggdf.FOD$IntFallPY = interaction(ggdf.FOD$Fallers, ggdf.FOD$PhysicalActivity, sep = " ")

### rIFG rSTN -----
boxplot = ggplot(data = ggdf.FOD, aes(x = IntFallPY, y = rIFG.31.FOD, fill = Fallers)) +
  geom_violinhalf(position = position_nudge(x = -.175, y = 0), flip = TRUE) +
  geom_jitter(width = 0.1) +
  geom_boxplot(width = 0.1, position = position_nudge(x = 0.25, y = 0),
  outlier.alpha = 0) + # fill = "white", alpha = 0.7, color = "grey") +
  # geom_flat_violin(position = position_nudge(x = .2, y = 0)) +
  # geom_jitter(alpha = 0.01, width = 0.15) +
  # facet_wrap(. ~ PhysicalActivity) +
  xlab(label = "Fall and Activity Levels") +
  ylab(label = "AFD (z-Score)") +
  labs(title = "rIFG to rSTN: AFD Value by Fall Group") +
  theme_light() +
  theme(axis.text.x = element_text(size = 8, angle = 25, hjust = 1)) +
  # Add p vals
  geom_bracket(inherit.aes = FALSE,
  xmin = "Not Faller Low Activity", xmax = "Faller Low Activity", y.position =
-3.05, tip.length = -0.015, vjust = 2.25,
  label = paste("****"),
  family = "Times") +
  geom_bracket(inherit.aes = FALSE,
  xmin = "Not Faller Moderate Activity", xmax = "Faller Moderate Activity",
y.position = -2.75, tip.length = -0.015, vjust = 2.25,
  label = paste("ns."),
  family = "Times") +
  geom_bracket(inherit.aes = FALSE,
  xmin = "Not Faller High Activity", xmax = "Faller High Activity", y.position
= -2.75, tip.length = -0.015, vjust = 2.25,
  label = paste("ns."),
  family = "Times") +
  scale_fill_viridis_d(alpha = 0.8) +
  theme(legend.position = "none", text = element_text(family = "Times"))
boxplot
ggsave(filename = paste(save_int_path, "rIFG_AFD_dotplot.pdf", sep = ""), width = 7, height
= 5)

# Alright investigate differences!
H1.LA = IndependantTest(ggdf.FOD$rIFG.31.FOD[ggdf.FOD$IntFallPY == "Not Faller Low
Activity"],
  ggdf.FOD$rIFG.31.FOD[ggdf.FOD$IntFallPY == "Faller Low Activity"],
  testName = "AFD Values Low Activity: Faller vs NotFaller",
  directionality = "two.sided")
capture.output(H1.LA, file = paste(save_int_path, "rIFG_H1-LA.txt", sep = ""))
H1.MA = IndependantTest(ggdf.FOD$rIFG.31.FOD[ggdf.FOD$IntFallPY == "Not Faller Moderate
Activity"],
  ggdf.FOD$rIFG.31.FOD[ggdf.FOD$IntFallPY == "Faller Moderate
Activity"],

```

```

        testName = "AFD Values Low Activity: Faller vs NotFaller",
        directionality = "two.sided")
capture.output(H1.MA, file = paste(save_int_path, "rIFG_H1-MA.txt", sep = ""))
H1.HA = IndependentTest(ggdf.FOD$rIFG.31.FOD[ggdf.FOD$IntFallPY == "Not Faller High
Activity"],
        ggdf.FOD$rIFG.31.FOD[ggdf.FOD$IntFallPY == "Faller High Activity"],
        testName = "AFD Values Low Activity: Faller vs NotFaller",
        directionality = "two.sided")
capture.output(H1.HA, file = paste(save_int_path, "rIFG_H1-HA.txt", sep = ""))
# Not fallers are all insignificant to one another
# Fallers are except low to moderate
# H1.NF.LAMA = IndependentTest(ggdf.FOD$rIFG.31.FOD[ggdf.FOD$IntFallPY == "Faller Moderate
Activity"],
#
#                                     ggdf.FOD$rIFG.31.FOD[ggdf.FOD$IntFallPY == "Faller
Low Activity"],
#
#                                     testName = "AFD Values NotFaller: L vs H Activity",
#                                     directionality = "two.sided")

# linear plot
linearplot = ggplot(data = ggdf.FOD, aes(x = Age, y = rIFG.31.FOD, color = Fallers, fill =
Fallers)) +
  geom_jitter() +
  theme_light() +
  facet_wrap(PhysicalActivity ~ ., nrow = 1) +
  geom_smooth(method = lm, alpha = 0.2) + theme(text = element_text(family = "Times")) +
  theme(text = element_text(family = "Times")) +
  xlab(label = "Age") +
  ylab(label = "AFD (z-Score)") +
  labs(title = "rIFG to rSTN: AFD Value by Age", color = "Category", fill = "Category") +
  scale_color_viridis_d(alpha = 0.8) +
  scale_fill_viridis_d(alpha = 0.8)
linearplot
ggsave(filename = paste(save_int_path, "rIFG_AFD_linplot.pdf", sep = ""), width = 10, height
= 3.5)

### 68 - rSTN -----

#### Set up df -----
# exclude above 3
ggdf.FOD.68 = df
ggdf.FOD.68$FOD.outliers = ID.outliers(ggdf.FOD.68$x68_31.FOD, 3)
ggdf.FOD.68 = data.frame(df$fallers_w3, df$x68_31.FOD, df$age_W3, df$BP_W3,
        ggdf.FOD.68$FOD.outliers, df$FRexercise3_W3)
colnames(ggdf.FOD.68) = c("Fallers", "x68.31.FOD", "Age", "BloodPressure", "Outliers",
"PhysicalActivity")
ggdf.FOD.68 = ggdf.FOD.68[!is.na(ggdf.FOD.68$x68.31.FOD),]
ggdf.FOD.68 = ggdf.FOD.68[!ggdf.FOD.68$Outliers, ]
ggdf.FOD.68$Fallers = factor(ggdf.FOD.68$Fallers, levels = c(0, 1), labels = c("Not
Faller", "Faller"))
ggdf.FOD.68 = ggdf.FOD.68[!is.na(ggdf.FOD.68$PhysicalActivity), ]
ggdf.FOD.68$IntFallPY = interaction(ggdf.FOD.68$Fallers, ggdf.FOD.68$PhysicalActivity, sep
= " ")

#### Plots -----
boxplot = ggplot(data = ggdf.FOD.68, aes(x = IntFallPY, y = x68.31.FOD, fill = Fallers)) +
  geom_violinhalf(position = position_nudge(x = -.175, y = 0), flip = TRUE) +
  geom_jitter(width = 0.1) +
  geom_boxplot(width = 0.1, position = position_nudge(x = 0.25, y = 0),
        outlier.alpha = 0) + # fill = "white", alpha = 0.7, color = "grey") +
  # geom_flat_violin(position = position_nudge(x = .2, y = 0)) +
  # geom_jitter(alpha = 0.01, width = 0.15) +
  # facet_wrap(. ~ PhysicalActivity) +
  xlab(label = "Fall and Actvity Levels") +
  ylab(label = "AFD (z-Score)") +
  labs(title = "rIFG(Subregion 68) to rSTN: AFD Value by Fall Group") +
  theme_light() +
  theme(axis.text.x = element_text(size = 8, angle = 25, hjust = 1)) +

```

```

# Add p vals
geom_bracket(inherit.aes = FALSE,
  xmin = "Not Faller Low Activity", xmax = "Faller Low Activity", y.position =
-3.05, tip.length = -0.015, vjust = 2.25,
  label = paste("***"),
  family = "Times") +
geom_bracket(inherit.aes = FALSE,
  xmin = "Not Faller Moderate Activity", xmax = "Faller Moderate Activity",
y.position = -2.75, tip.length = -0.015, vjust = 2.25,
  label = paste("ns."),
  family = "Times") +
geom_bracket(inherit.aes = FALSE,
  xmin = "Not Faller High Activity", xmax = "Faller High Activity", y.position
= -2.75, tip.length = -0.015, vjust = 2.25,
  label = paste("ns."),
  family = "Times") +
scale_fill_viridis_d(alpha = 0.8) +
theme(legend.position = "none", text = element_text(family = "Times"))
boxplot
ggsave(filename = paste(save_int_path, "x68_AFD_dotplot.pdf", sep = ""), width = 7, height =
5)

# Alright investigate differences!
H1.LA = IndependentTest(ggdf.FOD.68$x68.31.FOD[ggdf.FOD.68$IntFallPY == "Not Faller Low
Activity"],
  ggdf.FOD.68$x68.31.FOD[ggdf.FOD.68$IntFallPY == "Faller Low
Activity"],
  testName = "AFD Values Low Activity: Faller vs NotFaller",
  directionality = "two.sided")
capture.output(H1.LA, file = paste(save_int_path, "x68_H1-LA.txt", sep = ""))
H1.MA = IndependentTest(ggdf.FOD.68$x68.31.FOD[ggdf.FOD.68$IntFallPY == "Not Faller
Moderate Activity"],
  ggdf.FOD.68$x68.31.FOD[ggdf.FOD.68$IntFallPY == "Faller Moderate
Activity"],
  testName = "AFD Values Low Activity: Faller vs NotFaller",
  directionality = "two.sided")
capture.output(H1.MA, file = paste(save_int_path, "x68_H1-MA.txt", sep = ""))
H1.HA = IndependentTest(ggdf.FOD.68$x68.31.FOD[ggdf.FOD.68$IntFallPY == "Not Faller High
Activity"],
  ggdf.FOD.68$x68.31.FOD[ggdf.FOD.68$IntFallPY == "Faller High
Activity"],
  testName = "AFD Values Low Activity: Faller vs NotFaller",
  directionality = "two.sided")
capture.output(H1.HA, file = paste(save_int_path, "x68_H1-HA.txt", sep = ""))
# Not fallers are all insignificant to one another
# Fallers are except low to moderate

# linear plot
linearplot = ggplot(data = ggdf.FOD.68, aes(x = Age, y = x68.31.FOD, color = Fallers, fill =
Fallers)) +
  geom_jitter() +
  theme_light() +
  facet_wrap(PhysicalActivity ~ ., nrow = 1) +
  geom_smooth(method = lm, alpha = 0.2) + theme(text = element_text(family = "Times")) +
  theme(text = element_text(family = "Times")) +
  xlab(label = "Age") +
  ylab(label = "AFD (z-Score)") +
  labs(title = "rIFG(Subregion 68) to rSTN: AFD Value by Age", color = "Category", fill =
"Category") +
  scale_color_viridis_d(alpha = 0.8) +
  scale_fill_viridis_d(alpha = 0.8)
linearplot
ggsave(filename = paste(save_int_path, "x68_AFD_linplot.pdf", sep = ""), width = 10, height
= 3.5)

# ### Investigate BP -----
# # linear plot

```

```

# linearplot = ggplot(data = ggdf.FOD, aes(x = Age, y = rIFG.31.FOD, color = Fallers, fill =
Fallers)) +
#   geom_jitter() +
#   theme_light() +
#   facet_wrap(. ~ BloodPressure) +
#   geom_smooth(method = lm, alpha = 0.2) +   theme(text = element_text(family = "Times"))
+
#   theme(text = element_text(family = "Times")) +
#   xlab(label = "Age") +
#   ylab(label = "AFD (z-Score)") +
#   labs(title = "rIFG rSTN: AFD Value by Age", color = "Category", fill = "Category") +
#   scale_color_viridis_d(alpha = 0.8) +
#   scale_fill_viridis_d(alpha = 0.8)
#
#
# linearplot
# ggsave(filename = paste(save_path, "_int_BP_AFD_linplot.pdf", sep = ""), width = 5, height
= 4)
#
# ### New Plots -----
# # new df
# # exclude above 3
# ggdf.FA = df
# ggdf.FA$FA.outliers = ID.outliers(ggdf.FA$rIFG_31.FA, 3)
# ggdf.FA = ggdf.FA[!ggdf.FA$FA.outliers, ]
# ggdf.FA = ggdf.FA[!is.na(ggdf.FA$rIFG_31.FA),]
#
# ggdf.FA = data.frame(ggdf.FA$fallers_w3, ggdf.FA$rIFG_31.FA, ggdf.FA$age_W3)
# colnames(ggdf.FA) = c("Fallers", "rIFG.31.FA", "Age")
# ggdf.FA$Fallers = factor(ggdf.FA$Fallers, levels = c(0, 1), labels = c("Not Faller",
"Faller"))
#
# # Boxplot with FA
# boxplot = ggplot(data = ggdf.FA, aes(x = Fallers, y = rIFG.31.FA, fill = Fallers)) +
#   geom_violinhalf(position = position_nudge(x = .175, y = 0)) +
#   geom_jitter(width = 0.1) +
#   geom_boxplot(width = 0.1, position = position_nudge(x = -0.2, y = 0),
#                 outlier.alpha = 0) + # fill = "white", alpha = 0.7, color = "grey") +
#   # geom_flat_violin(position = position_nudge(x = .2, y = 0)) +
#   # geom_jitter(alpha = 0.01, width = 0.15) +
#   # facet_wrap(. ~ BloodPressure) +
#   xlab(label = "Fall") +
#   ylab(label = "FA (z-Score)") +
#   labs(title = "rIFG rSTN: FA Value by Fall Group") +
#   theme_light() +
#   scale_fill_viridis_d(alpha = 0.8) +
#   theme(legend.position = "none", text = element_text(family = "Times"))
# boxplot
# ggsave(filename = paste(save_path, "_FA_dotplot.pdf", sep = ""), width = 5, height = 4)
#
# # Dotplot with FA
# linearplot = ggplot(data = ggdf.FA, aes(x = Age, y = rIFG.31.FA, color = Fallers, fill =
Fallers)) +
#   geom_jitter() +
#   theme_light() +
#   # facet_wrap(. ~ BloodPressure) +
#   geom_smooth(method = lm) +   theme(text = element_text(family = "Times")) +
#   xlab(label = "Age") +
#   ylab(label = "FA (z-Score)") +
#   labs(title = "rIFG rSTN: FA Value by Age", color = "Category", fill = "Category") +
#   scale_color_viridis_d(alpha = 0.8) +
#   scale_fill_viridis_d(alpha = 0.8)
#
# linearplot
# ggsave(filename = paste(save_path, "_FA_linplot.pdf", sep = ""), width = 5, height = 4)
#
# # 65 cutoff
# ggdf.FOD.onlyold = ggdf.FOD[ggdf.FOD$Age > 64,]

```

```

# ggdf.FA.onlyold = ggdf.FA[ggdf.FA$Age > 64,]
#
# # Dotplot with FOD only Old
# linearplot = ggplot(data = ggdf.FOD.onlyold, aes(x = Age, y = rIFG.31.FOD, color =
Fallers, fill = Fallers)) +
#   geom_jitter() +
#   theme_light() +
#   # facet_wrap(. ~ BloodPressure) +
#   geom_smooth(method = lm) +   theme(text = element_text(family = "Times")) +
#   xlab(label = "Age") +
#   ylab(label = "AFD (z-Score)") +
#   labs(title = "rIFG rSTN: AFD Value by Age (65+)", color = "Category", fill = "Category")
+
#   scale_color_viridis_d(alpha = 0.8) +
#   scale_fill_viridis_d(alpha = 0.8)
#
# linearplot
# ggsave(filename = paste(save_path, "_AFD_onlyOld(65)_linplot.pdf", sep = ""), width = 5,
height = 4)
#
# # Dotplot with AFD only Old
# linearplot = ggplot(data = ggdf.FA.onlyold, aes(x = Age, y = rIFG.31.FA, color = Fallers,
fill = Fallers)) +
#   geom_jitter() +
#   theme_light() +
#   # facet_wrap(. ~ BloodPressure) +
#   geom_smooth(method = lm) +   theme(text = element_text(family = "Times")) +
#   xlab(label = "Age") +
#   ylab(label = "FA (z-Score)") +
#   labs(title = "rIFG rSTN: FA Value by Age (65+)", color = "Category", fill = "Category")
+
#   scale_color_viridis_d(alpha = 0.8) +
#   scale_fill_viridis_d(alpha = 0.8)
#
# linearplot
# ggsave(filename = paste(save_path, "_FA_onlyOld(65)_linplot.pdf", sep = ""), width = 5,
height = 4)
#
# # boxplots with only 65 +
# boxplot = ggplot(data = ggdf.FOD.onlyold, aes(x = Fallers, y = rIFG.31.FOD, fill =
Fallers)) +
#   geom_violinhalf(position = position_nudge(x = .175, y = 0)) +
#   geom_jitter(width = 0.1) +
#   geom_boxplot(width = 0.1, position = position_nudge(x = -0.2, y = 0),
#                 outlier.alpha = 0) +
#   # facet_wrap(. ~ BloodPressure) +
#   xlab(label = "Fall") +
#   ylab(label = "AFD (z-Score)") +
#   labs(title = "rIFG rSTN: AFD Value by Fall Group") +
#   theme_light() +
#   scale_fill_viridis_d(alpha = 0.8) +
#   theme(legend.position = "none", text = element_text(family = "Times"))
# boxplot
# ggsave(filename = paste(save_path, "_AFD_onlyOld_dotplot.pdf", sep = ""), width = 5,
height = 4)
#
# # boxplots with only 65 +
# boxplot = ggplot(data = ggdf.FA.onlyold, aes(x = Fallers, y = rIFG.31.FA, fill = Fallers))
+
#   geom_violinhalf(position = position_nudge(x = .175, y = 0)) +
#   geom_jitter(width = 0.1) +
#   geom_boxplot(width = 0.1, position = position_nudge(x = -0.2, y = 0),
#                 outlier.alpha = 0) +
#   # facet_wrap(. ~ BloodPressure) +
#   xlab(label = "Fall") +
#   ylab(label = "FA (z-Score)") +
#   labs(title = "rIFG rSTN: FA Value by Fall Group") +
#   theme_light() +

```

```

# scale_fill_viridis_d(alpha = 0.8) +
# theme(legend.position = "none", text = element_text(family = "Times"))
# boxplot
# ggsave(filename = paste(save_path, "_FA_onlyOld_dotplot.pdf", sep = ""), width = 5, height
= 4)
#
#
# # Control Plots -----
# # exclude above 3
# ggdf.FOD = control
# ggdf.FOD$FOD.outliers = ID.outliers(ggdf.FOD$X238_31.FOD_PASS, 3)
# ggdf.FOD = ggdf.FOD[!ggdf.FOD$FOD.outliers, ]
# ggdf.FOD = ggdf.FOD[!is.na(ggdf.FOD$X238_31.FOD_PASS),]
# ggdf.FOD = data.frame(ggdf.FOD$fallers_w3, ggdf.FOD$X238_31.FOD_PASS, ggdf.FOD$age_W3,
ggdf.FOD$BP_W3, ggdf.FOD$DIS.impaired.cont)
#
# colnames(ggdf.FOD) = c("Fallers", "rFFA.rSTN.FOD", "Age", "BloodPressure", "Impairment")
# ggdf.FOD$Fallers = factor(ggdf.FOD$Fallers, levels = c(0, 1), labels = c("Not Faller",
"Faller"))
#
#
# ### Original Plots -----
# boxplot = ggplot(data = ggdf.FOD, aes(x = Fallers, y = rFFA.rSTN.FOD, fill = Fallers)) +
#   geom_violinhalf(position = position_nudge(x = .175, y = 0)) +
#   geom_jitter(width = 0.1) +
#   geom_boxplot(width = 0.1, position = position_nudge(x = -0.2, y = 0),
#               outlier.alpha = 0) + # fill = "white", alpha = 0.7, color = "grey") +
#   # geom_flat_violin(position = position_nudge(x = .2, y = 0)) +
#   # geom_jitter(alpha = 0.01, width = 0.15) +
#   # facet_wrap(. ~ BloodPressure) +
#   xlab(label = "Fall") +
#   ylab(label = "AFD (z-Score)") +
#   labs(title = "rFFA rSTN: AFD Value by Fall Group") +
#   theme_light() +
#   scale_fill_viridis_d(alpha = 0.8) +
#   theme(legend.position = "none", text = element_text(family = "Times"))
# boxplot
# ggsave(filename = paste(save_control_path, "_AFD_dotplot.pdf", sep = ""), width = 5,
height = 4)
#
# # linear plot
# linearplot = ggplot(data = ggdf.FOD, aes(x = Age, y = rFFA.rSTN.FOD, color = Fallers, fill
= Fallers)) +
#   geom_jitter() +
#   theme_light() +
#   facet_wrap(. ~ BloodPressure) +
#   geom_smooth(method = lm, level = .95, alpha = 0.2) + theme(text = element_text(family
= "Times")) +
#   xlab(label = "Age") +
#   ylab(label = "AFD (z-Score)") +
#   labs(title = "rFFA rSTN: AFD Value by Age", color = "Category", fill = "Category") +
#   scale_color_viridis_d(alpha = 0.8) +
#   scale_fill_viridis_d(alpha = 0.8)
#
# linearplot
# ggsave(filename = paste(save_control_path, "_BP_AFD_linplot.pdf", sep = ""), width = 5,
height = 4)
#
# ### New Plots -----
# # new control
# # exclude above 3
# ggdf.FA = control
# ggdf.FA$FA.outliers = ID.outliers(ggdf.FA$X238_31.FA_PASS, 3)
# ggdf.FA = ggdf.FA[!ggdf.FA$FA.outliers, ]
# ggdf.FA = ggdf.FA[!is.na(ggdf.FA$X238_31.FA_PASS),]
#
# ggdf.FA = data.frame(ggdf.FA$fallers_w3, ggdf.FA$X238_31.FA_PASS, ggdf.FA$age_W3)
# colnames(ggdf.FA) = c("Fallers", "rFFA.rSTN.FA", "Age")

```

```

# ggdf.FA$Fallers = factor(ggdf.FA$Fallers, levels = c(0, 1), labels = c("Not Faller",
"Faller"))
#
# # Boxplot with FA
# boxplot = ggplot(data = ggdf.FA, aes(x = Fallers, y = rFFA.rSTN.FA, fill = Fallers)) +
#   geom_violinhalf(position = position_nudge(x = .175, y = 0)) +
#   geom_jitter(width = 0.1) +
#   geom_boxplot(width = 0.1, position = position_nudge(x = -0.2, y = 0),
#     outlier.alpha = 0) + # fill = "white", alpha = 0.7, color = "grey") +
#   # geom_flat_violin(position = position_nudge(x = .2, y = 0)) +
#   # geom_jitter(alpha = 0.01, width = 0.15) +
#   # facet_wrap(. ~ BloodPressure) +
#   xlab(label = "Fall") +
#   ylab(label = "FA (z-Score)") +
#   labs(title = "rFFA rSTN: FA Value by Fall Group") +
#   theme_light() +
#   scale_fill_viridis_d(alpha = 0.8) +
#   theme(legend.position = "none", text = element_text(family = "Times"))
# boxplot
# ggsave(filename = paste(save_control_path, "_FA_dotplot.pdf", sep = ""), width = 5, height
= 4)
#
# # Dotplot with FA
# linearplot = ggplot(data = ggdf.FA, aes(x = Age, y = rFFA.rSTN.FA, color = Fallers, fill =
Fallers)) +
#   geom_jitter() +
#   theme_light() +
#   # facet_wrap(. ~ BloodPressure) +
#   geom_smooth(method = lm) + theme(text = element_text(family = "Times")) +
#   xlab(label = "Age") +
#   ylab(label = "FA (z-Score)") +
#   labs(title = "rFFA rSTN: FA Value by Age", color = "Category", fill = "Category") +
#   scale_color_viridis_d(alpha = 0.8) +
#   scale_fill_viridis_d(alpha = 0.8)
#
# linearplot
# ggsave(filename = paste(save_control_path, "_FA_linplot.pdf", sep = ""), width = 5, height
= 4)
#
# # 65 cutoff
# ggdf.FOD.onlyold = ggdf.FOD[ggdf.FOD$Age > 64,]
# ggdf.FA.onlyold = ggdf.FA[ggdf.FA$Age > 64,]
#
# # Dotplot with FOD only Old
# linearplot = ggplot(data = ggdf.FOD.onlyold, aes(x = Age, y = rFFA.rSTN.FOD, color =
Fallers, fill = Fallers)) +
#   geom_jitter() +
#   theme_light() +
#   # facet_wrap(. ~ BloodPressure) +
#   geom_smooth(method = lm) + theme(text = element_text(family = "Times")) +
#   xlab(label = "Age") +
#   ylab(label = "AFD (z-Score)") +
#   labs(title = "rFFA rSTN: AFD Value by Age (65+)", color = "Category", fill = "Category")
+
#   scale_color_viridis_d(alpha = 0.8) +
#   scale_fill_viridis_d(alpha = 0.8)
#
# linearplot
# ggsave(filename = paste(save_control_path, "_AFD_onlyOld(65)_linplot.pdf", sep = ""),
width = 5, height = 4)
#
# # Dotplot with FOD only Old
# linearplot = ggplot(data = ggdf.FA.onlyold, aes(x = Age, y = rFFA.rSTN.FA, color =
Fallers, fill = Fallers)) +
#   geom_jitter() +
#   theme_light() +
#   # facet_wrap(. ~ BloodPressure) +
#   geom_smooth(method = lm) + theme(text = element_text(family = "Times")) +

```



```

#   xlab(label = "Age") +
#   ylab(label = "FA (z-Score)") +
#   labs(title = "rFFA rSTN: FA Value by Age (65+)", color = "Category", fill = "Category")
+
#   scale_color_viridis_d(alpha = 0.8) +
#   scale_fill_viridis_d(alpha = 0.8)
#
# linearplot
# ggsave(filename = paste(save_control_path, "_FA_onlyOld(65)_linplot.pdf", sep = ""), width
= 5, height = 4)
#
# # boxplots with only 65 +
# boxplot = ggplot(data = ggdf.FOD.onlyold, aes(x = Fallers, y = rFFA.rSTN.FOD, fill =
Fallers)) +
#   geom_violinhalf(position = position_nudge(x = .175, y = 0)) +
#   geom_jitter(width = 0.1) +
#   geom_boxplot(width = 0.1, position = position_nudge(x = -0.2, y = 0),
#                 outlier.alpha = 0) +
#   # facet_wrap(. ~ BloodPressure) +
#   xlab(label = "Fall") +
#   ylab(label = "AFD (z-Score)") +
#   labs(title = "rFFA rSTN: AFD Value by Fall Group") +
#   theme_light() +
#   scale_fill_viridis_d(alpha = 0.8) +
#   theme(legend.position = "none", text = element_text(family = "Times"))
# boxplot
# ggsave(filename = paste(save_control_path, "_AFD_onlyOld_dotplot.pdf", sep = ""), width =
5, height = 4)
#
# # boxplots with only 65 +
# boxplot = ggplot(data = ggdf.FA.onlyold, aes(x = Fallers, y = rFFA.rSTN.FA, fill =
Fallers)) +
#   geom_violinhalf(position = position_nudge(x = .175, y = 0)) +
#   geom_jitter(width = 0.1) +
#   geom_boxplot(width = 0.1, position = position_nudge(x = -0.2, y = 0),
#                 outlier.alpha = 0) +
#   # facet_wrap(. ~ BloodPressure) +
#   xlab(label = "Fall") +
#   ylab(label = "FA (z-Score)") +
#   labs(title = "rFFA rSTN: FA Value by Fall Group") +
#   theme_light() +
#   scale_fill_viridis_d(alpha = 0.8) +
#   theme(legend.position = "none", text = element_text(family = "Times"))
# boxplot
# ggsave(filename = paste(save_control_path, "_FA_onlyOld_dotplot.pdf", sep = ""), width =
5, height = 4)
#
# # Prediction Plots -----
# # exclude above 3
# plot.name = "lpreSMA lSTN"
# prediction = df
# ggdf.FOD = prediction
# ggdf.FOD$FOD.outliers = ID.outliers(ggdf.FOD$X58_182.FOD, 3)
# ggdf.FOD = ggdf.FOD[!ggdf.FOD$FOD.outliers, ]
# ggdf.FOD = ggdf.FOD[!is.na(ggdf.FOD$X58_182.FOD),]
# ggdf.FOD = data.frame(ggdf.FOD$fallers_after_w3, ggdf.FOD$X58_182.FOD, ggdf.FOD$age_W3,
#                       ggdf.FOD$NumberOfMeds, ggdf.FOD$sex_W3)
# colnames(ggdf.FOD) = c("Fallers", "lpreSMA.lSTN.FOD", "Age", "NumberOfMeds", "Sex")
# ggdf.FOD$Fallers = factor(ggdf.FOD$Fallers, levels = c(0, 1), labels = c("Not Faller",
"Faller"))
#
#
# ### Original Plots -----
# boxplot = ggplot(data = ggdf.FOD, aes(x = Fallers, y = lpreSMA.lSTN.FOD, fill = Fallers))
+
#   geom_violinhalf(position = position_nudge(x = .175, y = 0)) +
#   geom_jitter(width = 0.1) +
#   geom_boxplot(width = 0.1, position = position_nudge(x = -0.2, y = 0),

```

```

#         outlier.alpha = 0) + # fill = "white", alpha = 0.7, color = "grey") +
# # geom_flat_violin(position = position_nudge(x = .2, y = 0)) +
# # geom_jitter(alpha = 0.01, width = 0.15) +
# # facet_wrap(. ~ BloodPressure) +
# xlab(label = "Fall") +
# ylab(label = "AFD (z-Score)") +
# labs(title = paste(plot.name, "AFD Value by Fall Group")) +
# theme_light() +
# scale_fill_viridis_d(alpha = 0.8) +
# theme(legend.position = "none", text = element_text(family = "Times"))
# boxplot
# ggsave(filename = paste(save_prediction_path, "_", plot.name, "_AFD_dotplot.pdf", sep =
# ""), width = 5, height = 4)
#
# # linear plot
# linearplot = ggplot(data = ggdf.FOD, aes(x = Age, y = lpreSMA.lSTN.FOD, color = Fallers,
# fill = Fallers)) +
#   geom_jitter() +
#   theme_light() +
#   # facet_wrap(. ~ BloodPressure) +
#   geom_smooth(method = lm) + theme(text = element_text(family = "Times")) +
#   xlab(label = "Age") +
#   ylab(label = "AFD (z-Score)") +
#   labs(title = paste(plot.name, "AFD Value by Age"), color = "Category", fill =
# "Category") +
#   scale_color_viridis_d(alpha = 0.8) +
#   scale_fill_viridis_d(alpha = 0.8)
#
# linearplot
# ggsave(filename = paste(save_prediction_path, "_", plot.name, "_AFD_linplot.pdf", sep =
# ""), width = 5, height = 4)
#
#
# ##### Investigate Number of Meds -----
# # linear plot
# linearplot = ggplot(data = ggdf.FOD, aes(x = Age, y = lpreSMA.lSTN.FOD, color =
# NumberOfMeds)) +
#   geom_jitter(size = ((ggdf.FOD$NumberOfMeds/2) +2)) +
#   #shape = ifelse((as.logical(as.numeric(ggdf.FOD$Sex)-1)), 21, 24)) +
#   theme_light() +
#   facet_wrap(Sex ~ Fallers) +
#   geom_smooth(method = lm) +
#   theme(text = element_text(family = "Times")) +
#   xlab(label = "Age") +
#   ylab(label = "AFD (z-Score)") +
#   labs(title = paste(plot.name, "AFD Value by Age"), color = "Number of Meds") +
#   scale_color_viridis_c(alpha = 0.8) +
#   scale_fill_viridis_c(alpha = 0.8)
#
# linearplot
# ggsave(filename = paste(save_prediction_path, "_", plot.name,
# "_pred_invest_vars_linplot.pdf", sep = ""), width = 10, height = 8)
#
#
# ### New Plots -----
# # new prediction
# # exclude above 3
# ggdf.FA = prediction
# ggdf.FA$FA.outliers = ID.outliers(ggdf.FA$X58_182.FA, 3)
# ggdf.FA = ggdf.FA[!ggdf.FA$FA.outliers, ]
# ggdf.FA = ggdf.FA[!is.na(ggdf.FA$X58_182.FA),]
# ggdf.FA = data.frame(ggdf.FA$fallers_after_w3, ggdf.FA$X58_182.FA, ggdf.FA$age_W3)
# colnames(ggdf.FA) = c("Fallers", "lpreSMA.lSTN.FA", "Age")
# ggdf.FA$Fallers = factor(ggdf.FA$Fallers, levels = c(0, 1), labels = c("Not Faller",
# "Faller"))
#
# # Boxplot with FA
# boxplot = ggplot(data = ggdf.FA, aes(x = Fallers, y = lpreSMA.lSTN.FA, fill = Fallers)) +

```

```

# geom_violinhalf(position = position_nudge(x = .175, y = 0)) +
# geom_jitter(width = 0.1) +
# geom_boxplot(width = 0.1, position = position_nudge(x = -0.2, y = 0),
#             outlier.alpha = 0) + # fill = "white", alpha = 0.7, color = "grey") +
# # geom_flat_violin(position = position_nudge(x = .2, y = 0)) +
# # geom_jitter(alpha = 0.01, width = 0.15) +
# # facet_wrap(. ~ BloodPressure) +
# xlab(label = "Fall") +
# ylab(label = "FA (z-Score)") +
# labs(title = paste(plot.name, "FA Value by Fall Group")) +
# theme_light() +
# scale_fill_viridis_d(alpha = 0.8) +
# theme(legend.position = "none", text = element_text(family = "Times"))
# boxplot
# ggsave(filename = paste(save_prediction_path, "-", plot.name, "_FA_dotplot.pdf", sep =
# ""), width = 5, height = 4)
#
# # Dotplot with FA
# linearplot = ggplot(data = ggdf.FA, aes(x = Age, y = lpreSMA.lSTN.FA, color = Fallers,
# fill = Fallers)) +
#   geom_jitter() +
#   theme_light() +
#   # facet_wrap(. ~ BloodPressure) +
#   geom_smooth(method = lm) + theme(text = element_text(family = "Times")) +
#   xlab(label = "Age") +
#   ylab(label = "FA (z-Score)") +
#   labs(title = paste(plot.name, "FA Value by Age"), color = "Category", fill = "Category")
#
#   scale_color_viridis_d(alpha = 0.8) +
#   scale_fill_viridis_d(alpha = 0.8)
#
# linearplot
# ggsave(filename = paste(save_prediction_path, "-", plot.name, "_FA_linplot.pdf", sep =
# ""), width = 5, height = 4)
#
# # 65 cutoff
# ggdf.FOD.onlyold = ggdf.FOD[ggdf.FOD$Age > 64,]
# ggdf.FA.onlyold = ggdf.FA[ggdf.FA$Age > 64,]
#
# # Dotplot with FOD only Old
# linearplot = ggplot(data = ggdf.FOD.onlyold, aes(x = Age, y = lpreSMA.lSTN.FOD, color =
# Fallers, fill = Fallers)) +
#   geom_jitter() +
#   theme_light() +
#   # facet_wrap(. ~ BloodPressure) +
#   geom_smooth(method = lm) + theme(text = element_text(family = "Times")) +
#   xlab(label = "Age") +
#   ylab(label = "AFD (z-Score)") +
#   labs(title = paste(plot.name, "AFD Value by Age (65+)"), color = "Category", fill =
# "Category") +
#   scale_color_viridis_d(alpha = 0.8) +
#   scale_fill_viridis_d(alpha = 0.8)
#
# linearplot
# ggsave(filename = paste(save_prediction_path, "-", plot.name,
# "_AFD_onlyOld(65)_linplot.pdf", sep = ""), width = 5, height = 4)
#
# # Dotplot with FOD only Old
# linearplot = ggplot(data = ggdf.FA.onlyold, aes(x = Age, y = lpreSMA.lSTN.FA, color =
# Fallers, fill = Fallers)) +
#   geom_jitter() +
#   theme_light() +
#   # facet_wrap(. ~ BloodPressure) +
#   geom_smooth(method = lm) + theme(text = element_text(family = "Times")) +
#   xlab(label = "Age") +
#   ylab(label = "FA (z-Score)") +
#   labs(title = paste(plot.name, "FA Value by Age (65+)"), color = "Category", fill =
# "Category") +

```

```

# scale_color_viridis_d(alpha = 0.8) +
# scale_fill_viridis_d(alpha = 0.8)
#
# linearplot
# ggsave(filename = paste(save_prediction_path, "_", plot.name,
# "_FA_onlyOld(65)_linplot.pdf", sep = ""), width = 5, height = 4)
#
# # boxplots with only 65 +
# boxplot = ggplot(data = ggdf.FOD.onlyold, aes(x = Fallers, y = lpreSMA.lSTN.FOD, fill =
Fallers)) +
#   geom_violinhalf(position = position_nudge(x = .175, y = 0)) +
#   geom_jitter(width = 0.1) +
#   geom_boxplot(width = 0.1, position = position_nudge(x = -0.2, y = 0),
#                 outlier.alpha = 0) +
#   # facet_wrap(. ~ BloodPressure) +
#   xlab(label = "Fall") +
#   ylab(label = "AFD (z-Score)") +
#   labs(title = paste(plot.name, "AFD Value by Fall Group")) +
#   theme_light() +
#   scale_fill_viridis_d(alpha = 0.8) +
#   theme(legend.position = "none", text = element_text(family = "Times"))
# boxplot
# ggsave(filename = paste(save_prediction_path, "_", plot.name, "_AFD_onlyOld_dotplot.pdf",
sep = ""), width = 5, height = 4)
#
# # boxplots with only 65 +
# boxplot = ggplot(data = ggdf.FA.onlyold, aes(x = Fallers, y = lpreSMA.lSTN.FA, fill =
Fallers)) +
#   geom_violinhalf(position = position_nudge(x = .175, y = 0)) +
#   geom_jitter(width = 0.1) +
#   geom_boxplot(width = 0.1, position = position_nudge(x = -0.2, y = 0),
#                 outlier.alpha = 0) +
#   # facet_wrap(. ~ BloodPressure) +
#   xlab(label = "Fall") +
#   ylab(label = "FA (z-Score)") +
#   labs(title = paste(plot.name, "FA Value by Fall Group")) +
#   theme_light() +
#   scale_fill_viridis_d(alpha = 0.8) +
#   theme(legend.position = "none", text = element_text(family = "Times"))
# boxplot
# ggsave(filename = paste(save_prediction_path, "_", plot.name, "_FA_onlyOld_dotplot.pdf",
sep = ""), width = 5, height = 4)
#
#
# # Control Prediction Plots -----
# # exclude above 3
# prediction = control
# ggdf.FOD = prediction
# ggdf.FOD$FOD.outliers = ID.outliers(ggdf.FOD$lIFG_182.FOD, 3)
# ggdf.FOD = ggdf.FOD[!ggdf.FOD$FOD.outliers, ]
# ggdf.FOD = data.frame(ggdf.FOD$fallers_w3, ggdf.FOD$lIFG_182.FOD, ggdf.FOD$age_W3)
# colnames(ggdf.FOD) = c("Fallers", "lIFG.lSTN.FOD", "Age")
# ggdf.FOD = ggdf.FOD[!is.na(ggdf.FOD$lIFG.lSTN.FOD),]
# ggdf.FOD$Fallers = factor(ggdf.FOD$Fallers, levels = c(0, 1), labels = c("Not Faller",
"Faller"))
#
# ### Original Plots -----
# boxplot = ggplot(data = ggdf.FOD, aes(x = Fallers, y = lIFG.lSTN.FOD, fill = Fallers)) +
#   geom_violinhalf(position = position_nudge(x = .175, y = 0)) +
#   geom_jitter(width = 0.1) +
#   geom_boxplot(width = 0.1, position = position_nudge(x = -0.2, y = 0),
#                 outlier.alpha = 0) + # fill = "white", alpha = 0.7, color = "grey" +
#   # geom_flat_violin(position = position_nudge(x = .2, y = 0)) +
#   # geom_jitter(alpha = 0.01, width = 0.15) +
#   # facet_wrap(. ~ BloodPressure) +
#   xlab(label = "Fall") +
#   ylab(label = "AFD (z-Score)") +
#   labs(title = "AFD Value by Fall Group") +

```

```

# theme_light() +
# scale_fill_viridis_d(alpha = 0.8) +
# theme(legend.position = "none", text = element_text(family = "Times"))
# boxplot
# ggsave(filename = paste(save_control_path, "_AFD_pred_dotplot.pdf", sep = ""), width = 5,
height = 4)
#
# # linear plot
# linearplot = ggplot(data = ggdf.FOD, aes(x = Age, y = lIFG.lSTN.FOD, color = Fallers, fill
= Fallers)) +
#   geom_jitter() +
#   theme_light() +
#   # facet_wrap(. ~ BloodPressure) +
#   geom_smooth(method = lm) +   theme(text = element_text(family = "Times")) +
#   xlab(label = "Age") +
#   ylab(label = "AFD (z-Score)") +
#   labs(title = "AFD Value by Age", color = "Category", fill = "Category") +
#   scale_color_viridis_d(alpha = 0.8) +
#   scale_fill_viridis_d(alpha = 0.8)
#
# linearplot
# ggsave(filename = paste(save_control_path, "_AFD_pred_linplot.pdf", sep = ""), width = 5,
height = 4)
#
# ### New Plots-----
# # new prediction
# # exclude above 3
# ggdf.FA = control
# ggdf.FA$FA.outliers = ID.outliers(ggdf.FA$lIFG_182.FOD, 3)
# ggdf.FA = ggdf.FA[!ggdf.FA$FA.outliers, ]
# ggdf.FA = ggdf.FA[!is.na(ggdf.FA$lIFG_182.FOD),]
#
# ggdf.FA = data.frame(ggdf.FA$fallers_w3, ggdf.FA$lIFG_182.FOD, ggdf.FA$age_W3)
# colnames(ggdf.FA) = c("Fallers", "lIFG.lSTN.FA", "Age")
# ggdf.FA$Fallers = factor(ggdf.FA$Fallers, levels = c(0, 1), labels = c("Not Faller",
"Faller"))
#
# # Boxplot with FA
# boxplot = ggplot(data = ggdf.FA, aes(x = Fallers, y = lIFG.lSTN.FA, fill = Fallers)) +
#   geom_violinhalf(position = position_nudge(x = .175, y = 0)) +
#   geom_jitter(width = 0.1) +
#   geom_boxplot(width = 0.1, position = position_nudge(x = -0.2, y = 0),
#                 outlier.alpha = 0) + # fill = "white", alpha = 0.7, color = "grey") +
#   # geom_flat_violin(position = position_nudge(x = .2, y = 0)) +
#   # geom_jitter(alpha = 0.01, width = 0.15) +
#   # facet_wrap(. ~ BloodPressure) +
#   xlab(label = "Fall") +
#   ylab(label = "FA (z-Score)") +
#   labs(title = "FA Value by Fall Group") +
#   theme_light() +
#   scale_fill_viridis_d(alpha = 0.8) +
#   theme(legend.position = "none", text = element_text(family = "Times"))
# boxplot
# ggsave(filename = paste(save_control_path, "_pred_FA_dotplot.pdf", sep = ""), width = 5,
height = 4)
#
# # Dotplot with FA
# linearplot = ggplot(data = ggdf.FA, aes(x = Age, y = lIFG.lSTN.FA, color = Fallers, fill =
Fallers)) +
#   geom_jitter() +
#   theme_light() +
#   # facet_wrap(. ~ BloodPressure) +
#   geom_smooth(method = lm) +   theme(text = element_text(family = "Times")) +
#   xlab(label = "Age") +
#   ylab(label = "FA (z-Score)") +
#   labs(title = "FA Value by Age", color = "Category", fill = "Category") +
#   scale_color_viridis_d(alpha = 0.8) +
#   scale_fill_viridis_d(alpha = 0.8)

```

```

#
# linearplot
# ggsave(filename = paste(save_control_path, "_pred_FA_linplot.pdf", sep = ""), width = 5,
height = 4)
#
# # 65 cutoff
# ggdf.FOD.onlyold = ggdf.FOD[ggdf.FOD$Age > 64,]
# ggdf.FA.onlyold = ggdf.FA[ggdf.FA$Age > 64,]
#
# # Dotplot with FOD only Old
# linearplot = ggplot(data = ggdf.FOD.onlyold, aes(x = Age, y = lIFG.lSTN.FOD, color =
Fallers, fill = Fallers)) +
#   geom_jitter() +
#   theme_light() +
#   # facet_wrap(. ~ BloodPressure) +
#   geom_smooth(method = lm) +   theme(text = element_text(family = "Times")) +
#   xlab(label = "Age") +
#   ylab(label = "AFD (z-Score)") +
#   labs(title = "AFD Value by Age (65+)", color = "Category", fill = "Category") +
#   scale_color_viridis_d(alpha = 0.8) +
#   scale_fill_viridis_d(alpha = 0.8)
#
# linearplot
# ggsave(filename = paste(save_control_path, "_AFD_pred_onlyOld(65)_linplot.pdf", sep = ""),
width = 5, height = 4)
#
# # Dotplot with FOD only Old
# linearplot = ggplot(data = ggdf.FA.onlyold, aes(x = Age, y = lIFG.lSTN.FA, color =
Fallers, fill = Fallers)) +
#   geom_jitter() +
#   theme_light() +
#   # facet_wrap(. ~ BloodPressure) +
#   geom_smooth(method = lm) +   theme(text = element_text(family = "Times")) +
#   xlab(label = "Age") +
#   ylab(label = "FA (z-Score)") +
#   labs(title = "FA Value by Age (65+)", color = "Category", fill = "Category") +
#   scale_color_viridis_d(alpha = 0.8) +
#   scale_fill_viridis_d(alpha = 0.8)
#
# linearplot
# ggsave(filename = paste(save_control_path, "_pred_FA_onlyOld(65)_linplot.pdf", sep = ""),
width = 5, height = 4)
#
# # boxplots with only 65 +
# boxplot = ggplot(data = ggdf.FOD.onlyold, aes(x = Fallers, y = lIFG.lSTN.FOD, fill =
Fallers)) +
#   geom_violinhalf(position = position_nudge(x = .175, y = 0)) +
#   geom_jitter(width = 0.1) +
#   geom_boxplot(width = 0.1, position = position_nudge(x = -0.2, y = 0),
#                 outlier.alpha = 0) +
#   # facet_wrap(. ~ BloodPressure) +
#   xlab(label = "Fall") +
#   ylab(label = "AFD (z-Score)") +
#   labs(title = "AFD Value by Fall Group") +
#   theme_light() +
#   scale_fill_viridis_d(alpha = 0.8) +
#   theme(legend.position = "none", text = element_text(family = "Times"))
# boxplot
# ggsave(filename = paste(save_control_path, "_AFD_pred_onlyOld_dotplot.pdf", sep = ""),
width = 5, height = 4)
#
# # boxplots with only 65 +
# boxplot = ggplot(data = ggdf.FA.onlyold, aes(x = Fallers, y = lIFG.lSTN.FA, fill =
Fallers)) +
#   geom_violinhalf(position = position_nudge(x = .175, y = 0)) +
#   geom_jitter(width = 0.1) +
#   geom_boxplot(width = 0.1, position = position_nudge(x = -0.2, y = 0),
#                 outlier.alpha = 0) +

```

```

# # facet_wrap(. ~ BloodPressure) +
# xlab(label = "Fall") +
# ylab(label = "FA (z-Score)") +
# labs(title = "FA Value by Fall Group") +
# theme_light() +
# scale_fill_viridis_d(alpha = 0.8) +
# theme(legend.position = "none", text = element_text(family = "Times"))
# boxplot
# ggsave(filename = paste(save_control_path, "_pred_FA_only0ld_dotplot.pdf", sep = ""),
width = 5, height = 4)

```

10.4.8 Classification Accuracy (R)

This script analyses most of the Classification Accuracy based Questions.

```
# This script we will Bring together the files
# Init -----
# clear all
rm(list = ls())

# Libraries
library(lmerTest)
library(lme4)
library(doParallel)
library(ggplot2)
library(extrafont)
library(ggpubr)
library(see)

# set wd
work_dir = ""
setwd(work_dir)

# get useful functions
source(file = "../.../Code/R Code/R_useful_funcs_share.R")

# Start Parallel core -----

# Load Data -----

# Load MEP Data
load(file = "")

# Load QS Data
load(file = "")

# dataframe with just up and down
df.updown = results[results$conditions %in% c("up", "down"),]

# exclude background rMS higher than 0.01
df.updown = df.updown[df.updown$actual_rMS_right < 0.01, ]

# ztransform rms, because it's small scale is causing issues
df.updown$ztrans.rms = (df.updown$actual_rMS_right -
mean(df.updown$actual_rMS_right))/sd(df.updown$actual_rMS_right)

# Remove Sess 1 and 2 from the EEG dataset
# Create group variable for CA scores:
TMS.PIDs = c('')
df.merged$Group = "EEG"
df.merged$Group[df.merged$PID %in% TMS.PIDs] = "TMS"
df.merged$Continuous.Block = (df.merged$Sess - 1) * 4 + df.merged$Block

df.merged$CA = as.numeric(df.merged$CA)
df.merged$`KVIQ Total` = as.numeric(df.merged$`KVIQ Total`)
df.merged$`KVIQ Imagery` = as.numeric(df.merged$`KVIQ Imagery`)
df.merged$`KVIQ Intensity` = as.numeric(df.merged$`KVIQ Intensity`)
df.merged$KVIQ.Intensity = df.merged$`KVIQ Intensity`

# Make succs hand and cross numeric
df.merged$`Success Cross` = as.numeric(df.merged$`Success Cross`)
df.merged$`Success Hand` = as.numeric(df.merged$`Success Hand`)
df.merged$`Dominant Hand Score` = as.numeric(df.merged$`Dominant Hand Score`)

# Make Age numeric
df.merged$Age = as.numeric(df.merged$Age)
```



```

df.merged$Sex = factor(df.merged$Sex, levels = c("f", "Female", "m", "Male"),
                        labels = c("Female", "Female", "Male", "Male"))

# H2 -----

H2.n0 = nullmodel_regression_function(df.eegbci, CA ~ 1, 'PID')
# so we should be using (1|PID)

# H2.m1 = add2model_regression_function(df.eegbci, CA ~ 1 + (1|PID), "Group",
# "Continuous.Block")
# H2.m2 = add2model_regression_function(df.eegbci, CA ~ 1 + (Group|PID), "Group",
# "Continuous.Block")

# PID|PID model:
# h.test = lmer(CA ~ Group * Continuous.Block + (PID|1), data = df.eegbci)

# compare two models
h2.m1.1 = lmer(CA ~ Group * Continuous.Block + (1|PID), data = df.eegbci)
save(h2.m1.1, file = "./Processed Data/CA Data/Model/mFull/mFull-model.rda")
write(capture.output(summary(h2.m1.1)), file = "./Processed Data/CA Data/Model/mFull/mFull-
model-output.txt")
h2.m2.1 = lmer(CA ~ Group * Continuous.Block + (Group|PID), data = df.eegbci)
save(h2.m2.1, file = "./Processed Data/CA Data/Model/mFull/mFull-grouprand-model.rda")
write(capture.output(summary(h2.m2.1)), file = "./Processed Data/CA Data/Model/mFull/mFull-
grouprand-model-output.txt")
write(capture.output(anova(h2.m1.1, h2.m2.1)), file = "./Processed Data/CA
Data/Model/mFull/mFull-gouprand-comp.txt")
# Group|PID not much better

# how about adding Mental imagery scores and kss?
# checking out kss
H2.m1.kviql.tot = lmer(CA ~ Group * Continuous.Block + `KVIQ Total` + (1|PID), data =
df.eegbci)
Save_lm_summ(H2.m1.kviql.tot, "./Processed Data/CA Data/Model/mKVIQ/mKVIQ-tot")
H2.m1.kviql.vis = lmer(CA ~ Group * Continuous.Block + `KVIQ Imagery` + (1|PID), data =
df.eegbci)
Save_lm_summ(H2.m1.kviql.vis, "./Processed Data/CA Data/Model/mKVIQ/mKVIQ-vis")
H2.m1.kviql.int = lmer(CA ~ Group * Continuous.Block + `KVIQ Intensity` + (1|PID), data =
df.eegbci)
Save_lm_summ(H2.m1.kviql.int, "./Processed Data/CA Data/Model/mKVIQ/mKVIQ-int")
summary(H2.m1.kviql.tot)
summary(H2.m1.kviql.vis)
summary(H2.m1.kviql.int)
anova(H2.m1.kviql.tot, H2.m1.kviql.vis, H2.m1.kviql.int)
write(capture.output(anova(H2.m1.kviql.tot, H2.m1.kviql.vis, H2.m1.kviql.int)), file =
"./Processed Data/CA Data/Model/mKVIQ/mKVIQ-comp.txt")

# H2.m1 = add2model_regression_function(df.eegbci, CA ~ 1 + Group * Continuous.Block +
(1|PID), "KSS", "KVIQ.Intensity")
# very interesting KSS makes model better! KVIQ not so much!
df.kss = df.eegbci[!is.na(df.eegbci$KSS),]
H2.m1.KSS = lmer(CA ~ Group * Continuous.Block + KSS + (1|PID), data = df.kss)
Save_lm_summ(H2.m1.KSS, "./Processed Data/CA Data/Model/mKSS/mKSS-KSS")
H2.m1.noKSS = lmer(CA ~ Group * Continuous.Block + (1|PID), data = df.kss)
Save_lm_summ(H2.m1.noKSS, "./Processed Data/CA Data/Model/mKSS/mKSS-noKSS")
H2.m1.Sess = lmer(CA ~ Group * Continuous.Block + Sess + (1|PID), data = df.kss)
Save_lm_summ(H2.m1.Sess, "./Processed Data/CA Data/Model/mKSS/mKSS-Sess")
H2.m1.KSSGroupInt = lmer(CA ~ Group * Continuous.Block + KSS + KSS:Group + (1|PID), data =
df.kss)
Save_lm_summ(H2.m1.KSSGroupInt, "./Processed Data/CA Data/Model/mKSS/mKSS-KSSGroupInt")
H2.m1.KSSGroupIntComp = lmer(CA ~ Group * Continuous.Block * KSS + (1|PID), data = df.kss)
Save_lm_summ(H2.m1.KSSGroupIntComp, "./Processed Data/CA Data/Model/mKSS/mKSS-
KSSGroupIntComp")
summary(H2.m1.KSS)
summary(H2.m1.noKSS)
summary(H2.m1.Sess)
summary(H2.m1.KSSGroupInt)
summary(H2.m1.KSSGroupIntComp)

```

```

anova(H2.m1.KSSGroupIntComp, H2.m1.KSS, H2.m1.noKSS, H2.m1.Sess, H2.m1.KSSGroupInt)
write(capture.output(anova(H2.m1.KSSGroupIntComp, H2.m1.KSS, H2.m1.noKSS, H2.m1.Sess,
H2.m1.KSSGroupInt)), file = "./Processed Data/CA Data/Model/mKSS/mKSS-comp.txt")

# just looking at failure bias for fun
FB.m1 = lmer(`Failure Bias` ~ Group * Continuous.Block + (1|PID), data = df.merged)
summary(FB.m1)
# interesting, training is a predictor of failure bias (succHand - succCross)
# this means that participants over time got better at hand or worse at cross
Succs.Hand.m1 = lmer(`Success Hand` ~ Group * Continuous.Block + (1|PID), data = df.merged)
Succs.Cross.m1 = lmer(`Success Cross` ~ Group * Continuous.Block + (1|PID), data =
df.merged)
summary(Succs.Hand.m1)
summary(Succs.Cross.m1)
Save_lm_summ(Succs.Hand.m1, "./Processed Data/CA Data/Model/mHand/mHand-Hand")
Save_lm_summ(Succs.Cross.m1, "./Processed Data/CA Data/Model/mHand/mHand-Cross")
Save_lm_summ(FB.m1, "./Processed Data/CA Data/Model/mHand/mHand-FailureBias")
anova(FB.m1, Succs.Hand.m1, Succs.Cross.m1)
# it does seem that TMS makes you (intitially) worse at hand uscces, but then over time
better
# same time it makes slightly (not signi) worse at cross

# Other things to look at: Sex and Age
SA.m1 = lmer(CA ~ Sex + Age + Group * Continuous.Block + Sess + (1|PID), data = df.merged)
summary(SA.m1)
Save_lm_summ(SA.m1, "./Processed Data/CA Data/Model/mOther/mOther-AgeSex")
# Sex and Age no influence

# Other things to look at:
DH.m1 = lmer(CA ~ `Dominant Hand Score` + Group * Continuous.Block + Sess + (1|PID), data =
df.merged)
summary(DH.m1)
Save_lm_summ(DH.m1, "./Processed Data/CA Data/Model/mOther/mOther-DomHand")
# Hand dominance score no influence

## H2 Plot -----
gg.plot = ggplot(data = df.eegbci, aes(x = Continuous.Block, y = CA, color = Group)) +
  geom_jitter(aes(size = (KSS/3 + 1), color = Group)) +
  facet_wrap(Group ~ .) +
  geom_smooth(method = lm, alpha = 0.2) +
  theme_light() +
  xlab(label = "Block Number Continuous") +
  ylab(label = "Classification Accuracy [%]") +
  labs(title = "TMS priming EEG-BCI use", color = "Group", size = "KSS") +
  theme(text = element_text(family = "Times")) +
  scale_color_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_fill_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8)
gg.plot
ggsave(filename = './Processed Data/CA Data/Plots/CA-group-linplot.pdf', width = 8, height =
4)

# H2.1 -----

# HYPOTHESIS 2.1: First EEG session
EEG.Sess1.CA = na.omit(df.merged$CA[(df.merged$Group == "EEG" & df.merged$Sess == 1)])
TMS.Sess3.CA = na.omit(df.eegbci$CA[(df.eegbci$Group == "TMS" & df.eegbci$Sess == 3)])

H2.1.ittd.old = IndependantTest(EEG.Sess1.CA, TMS.Sess3.CA, "First EEG Session Comparison,
EEG vs TMS", directionality = "less")
write(capture.output(H2.1.ittd.old), file = './Processed Data/CA Data/H2.1/H2-1-ittd-old.txt')
# this returns a wilcoxon rank sum test with continuity correction, but they look normally
distributed, so we're doing a t-test
H2.1.ittd = t.test(TMS.Sess3.CA, EEG.Sess1.CA, alternative = "g")
write(capture.output(H2.1.ittd), file = './Processed Data/CA Data/H2.1/H2-1-ittd.txt')

# dataframe for boxplot
EEG = data.frame(EEG.Sess1.CA)

```

```

EEG$Group = "EEG"
colnames(EEG) = c('CA', 'Group')

TMS = data.frame(TMS.Sess3.CA)
TMS$Group = "TMS"
colnames(TMS) = c('CA', 'Group')

ggdf.H2.1 = data.frame(rbind(EEG, TMS))

# boxplot for that
gg.plot = ggplot(ggdf.H2.1, aes(x = Group, y = CA, fill = Group)) +
  geom_violinhalf(position = position_nudge(x = -.075, y = 0), flip = TRUE) +
  geom_jitter(width = 0.04) +
  geom_boxplot(width = 0.1, position = position_nudge(x = 0.15, y = 0),
    outlier.alpha = 0) +
  theme_light() +
  xlab(label = "Group") +
  ylab(label = "Classification Accuracy [%]") +
  labs(title = "TMS priming EEG-BCI use, Comparison of first EEG-BCI Session", color =
"Group", size = "KSS") +
  theme(text = element_text(family = "Times")) +
  scale_color_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_fill_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  geom_bracket(inherit.aes = FALSE,
    xmin = "EEG", xmax = "TMS", y.position = 92, tip.length = 0.013,
    label = paste("p = 0.034"),
    family = "Times")

gg.plot
ggsave(filename = './Processed Data/CA Data/H2.1/H2-1-boxplot.pdf', width = 6, height = 6)

# H2.2 -----

EEG.Sess3.CA = na.omit(df.eegbci$CA[(df.eegbci$Group == "EEG" & df.eegbci$Sess == 3)])

H2.2.itt = IndependentTest(EEG.Sess3.CA, TMS.Sess3.CA, "Third Session Comparison, EEG vs
TMS", directionality = "l")
write(capture.output(H2.2.itt), file = './Processed Data/CA Data/H2.2/H2-2-itt.txt')

# dataframe for boxplot
EEG = data.frame(EEG.Sess3.CA)
EEG$Group = "EEG"
colnames(EEG) = c('CA', 'Group')

TMS = data.frame(TMS.Sess3.CA)
TMS$Group = "TMS"
colnames(TMS) = c('CA', 'Group')

ggdf.H2.2 = data.frame(rbind(EEG, TMS))

# boxplot for that
gg.plot = ggplot(ggdf.H2.2, aes(x = Group, y = CA, fill = Group)) +
  geom_violinhalf(position = position_nudge(x = -.075, y = 0), flip = TRUE) +
  geom_jitter(width = 0.04) +
  geom_boxplot(width = 0.1, position = position_nudge(x = 0.15, y = 0),
    outlier.alpha = 0) +
  theme_light() +
  xlab(label = "Group") +
  ylab(label = "Classification Accuracy [%]") +
  labs(title = "TMS priming EEG-BCI use Session 3", color = "Group", size = "KSS") +
  theme(text = element_text(family = "Times")) +
  scale_color_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_fill_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  geom_bracket(inherit.aes = FALSE,
    xmin = "EEG", xmax = "TMS", y.position = 95, tip.length = 0.013,
    label = paste("p = 0.77"),

```

```

family = "Times")

gg.plot
ggsave(filename = './Processed Data/CA Data/H2.2/H2-2-boxplot.pdf', width = 6, height = 6)

# H2.3 -----

EEG.Sess5.CA = na.omit(df.eegbci$CA[(df.eegbci$Group == "EEG" & df.eegbci$Sess == 5)])
TMS.Sess5.CA = na.omit(df.eegbci$CA[(df.eegbci$Group == "TMS" & df.eegbci$Sess == 5)])

H2.3.itt = IndependentTest(EEG.Sess5.CA, TMS.Sess5.CA, "Fifth Session Comparison, EEG vs
TMS", directionality = "less")
write(capture.output(H2.3.itt), file = './Processed Data/CA Data/H2.3/H2-3-itt.txt')

# dataframe for boxplot
EEG = data.frame(EEG.Sess5.CA)
EEG$Group = "EEG"
colnames(EEG) = c('CA', 'Group')

TMS = data.frame(TMS.Sess5.CA)
TMS$Group = "TMS"
colnames(TMS) = c('CA', 'Group')

ggdf.H2.3 = data.frame(rbind(EEG, TMS))

# boxplot for that
gg.plot = ggplot(ggdf.H2.3, aes(x = Group, y = CA, fill = Group)) +
  geom_violinhalf(position = position_nudge(x = -.075, y = 0), flip = TRUE) +
  geom_jitter(width = 0.04) +
  geom_boxplot(width = 0.1, position = position_nudge(x = 0.15, y = 0),
    outlier.alpha = 0) +
  theme_light() +
  xlab(label = "Group") +
  ylab(label = "Classification Accuracy [%]") +
  labs(title = "TMS priming EEG-BCI use Session 5", color = "Group", size = "KSS") +
  theme(text = element_text(family = "Times")) +
  scale_color_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_fill_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  geom_bracket(inherit.aes = FALSE,
    xmin = "EEG", xmax = "TMS", y.position = 102, tip.length = 0.013,
    label = paste("p = 0.034*"),
    family = "Times")

gg.plot
ggsave(filename = './Processed Data/CA Data/H2.3/H2-3-boxplot.pdf', width = 6, height = 6)

# H2.4 -----

# take all sess 1 (& 3) results
EEG.Sess1 = df.merged[(df.merged$Group == "EEG" & df.merged$Sess == 1 & df.merged$Block !=
0), c("PID", "Block", "CA")]
TMS.Sess3 = df.merged[(df.merged$Group == "TMS" & df.merged$Sess == 3 & df.merged$Block !=
0), c("PID", "Block", "CA")]
EEG.Sess1$slopeUID = interaction(EEG.Sess1$PID, EEG.Sess1$Block)
TMS.Sess3$slopeUID = interaction(TMS.Sess3$PID, TMS.Sess3$Block)

# take all sess 5 results
EEG.Sess5 = df.merged[(df.merged$Group == "EEG" & df.merged$Sess == 5 & df.merged$Block !=
0), c("PID", "Block", "CA")]
TMS.Sess5 = df.merged[(df.merged$Group == "TMS" & df.merged$Sess == 5 & df.merged$Block !=
0), c("PID", "Block", "CA")]
EEG.Sess5$slopeUID = interaction(EEG.Sess5$PID, EEG.Sess5$Block)
TMS.Sess5$slopeUID = interaction(TMS.Sess5$PID, TMS.Sess5$Block)

# Make paired within groups
EEG.Slope = EEG.Sess5$CA - EEG.Sess1$CA
TMS.Slope = TMS.Sess5$CA - TMS.Sess3$CA

```

```

# EEG.Slope = MakePaired(EEG.Sess1, EEG.Sess5, "slopeUID")
# TMS.Slope = MakePaired(TMS.Sess3, TMS.Sess5, "slopeUID")
# EEG.Slope$Slope = EEG.Slope$result$CA.x - EEG.Slope$result$CA.y
# TMS.Slope$Slope = TMS.Slope$result$CA.x - TMS.Slope$result$CA.y

# do ttest with diff.eeg, diff.tms
H2.4.res = IndependentTest(TMS.Slope, EEG.Slope, testName = "TMS vs EEG: Slope",
                           directionality = 'greater')
write(capture.output(H2.4.res), file = "../Processed Data/CA Data/H2.4/H2.4-indttest-
result.txt")

# plot just EEG CA's

gg.plot = ggplot(data = df.eegbci, aes(x = Continuous.Block, y = CA, color = Group)) +
  geom_jitter(size = 4) + # aes(size = (KSS/3 + 1) , color = Group)) +
  #facet_wrap(Group ~ .) +
  geom_smooth(method = lm, alpha = 0.2) +
  theme_light() +
  xlab(label = "Block Number Continuous") +
  ylab(label = "Classification Accuracy [%]") +
  labs(title = "TMS priming EEG-BCI use", color = "Group") +
  theme(text = element_text(family = "Times")) +
  scale_color_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_fill_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8)
gg.plot

gg.plot = ggplot(data = df.merged, aes(x = Continuous.Block, y = CA, color = Group)) +
  geom_jitter(size = 4) + # aes(size = (KSS/3 + 1) , color = Group)) +
  #facet_wrap(Group ~ .) +
  geom_smooth(method = lm, alpha = 0.2) +
  theme_light() +
  xlab(label = "Block Number Continuous") +
  ylab(label = "Classification Accuracy [%]") +
  labs(title = "TMS priming EEG-BCI use", color = "Group") +
  theme(text = element_text(family = "Times")) +
  scale_color_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_fill_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8)
gg.plot

# ggsave(filename = '../Processed Data/CA Data/Plots/CA-group-linplot.pdf', width = 8, height
= 4)

# KSS Investigation -----

KSS.Summary = list()
KSS.Summary$MeanTMS = mean(df.kss$KSS[df.kss$Group == "TMS"])
KSS.Summary$MeanEEG = mean(df.kss$KSS[df.kss$Group == "EEG"])
KSS.TMS = df.kss$KSS[df.kss$Group == "TMS"]
KSS.EEG = df.kss$KSS[df.kss$Group == "EEG"]
KSS.Summary$Test = IndependentTest(KSS.TMS, KSS.EEG, "KSS Scores: TMS v EEG")

# plot differences:
gg.plot = ggplot(data = df.kss, aes(x = KSS, y = CA, color = Group)) +
  geom_jitter(size = 4) + # aes(size = (KSS/3 + 1) , color = Group)) +
  facet_wrap(Group ~ ., nrow = 1) +
  geom_smooth(method = lm, alpha = 0.2) +
  theme_light() +
  xlab(label = "KSS [Increasing fatigue]") +
  ylab(label = "Classification Accuracy [%]") +
  labs(title = "Fatigue among the groups", color = "Group") +
  theme(text = element_text(family = "Times")) +
  scale_color_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_fill_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8)
gg.plot

# Linear model looking at fatigue by group:

```

```

Fat.full = lmer(KSS ~ Group * `Success Cross` + Group:`Success Hand` + `Success Hand` +
(1|PID), data = df.kss)
summary(Fat.full)
Fat.cross = lmer(KSS ~ Group * `Success Cross` + (1|PID), data = df.kss)
summary(Fat.cross)
Fat.hand = lmer(KSS ~ Group * `Success Hand` + (1|PID), data = df.kss)
summary(Fat.hand)
anova(Fat.full, Fat.cross, Fat.hand)

# good vs bad users
df.achievers = BCI_Achiever(df.merged.1)
save(df.achievers, file = "../Processed Data/CA Data/df-Achievers.R")
write.csv(df.achievers, file = "../Processed Data/QS Data/df-Achievers.csv")

```

10.4.9 Power 1 (R)

This Script makes the powerband data ready to create linear models.

```
# This script we will look at the Power Metrics

# Init -----
# clear all
rm(list = ls())

# Libraries
library(lmerTest)
library(lme4)
library(doParallel)
library(ggplot2)
library(extrafont)
library(ggpubr)
library(see)
library(readxl)

# set wd
work_dir = ""
setwd(work_dir)

# get useful functions
source(file = "../../../Code/R Code/R_useful_funcs_share.R")

# there are multiple files to look at here:
power.files = list.files(path = "../Processed Data/EEG Power/Data/", pattern = ".csv")

# read all of them in:
df.Pbs = list()
for (powerband in power.files) {
  powerbandname = strsplit(powerband, split = ".csv")[[1]]
  df.Pbs[[powerbandname]] = read.csv(paste0("../Processed Data/EEG Power/Data/",
powerbandname, ".csv"))
}

# Data Adjustements -----

# Continuous block numbers
for (lists in names(df.Pbs)) {
}
for (lists in names(df.Pbs)) {
  df.Pbs[[lists]]$Continuous.Block = (df.Pbs[[lists]]$Ses - 1) * 4 + df.Pbs[[lists]]$Block
  check_summary(df.Pbs[[lists]]$Continuous.Block)
}

# Delete any PID = 0
for (lists in names(df.Pbs)) {
  df.Pbs[[lists]] = df.Pbs[[lists]][!df.Pbs[[lists]]$Continuous.Block == -4, ]
  check_summary(df.Pbs[[lists]]$Continuous.Block)
}

# Add Groups
TMS.PIDs = c()
for (lists in names(df.Pbs)) {
  df.Pbs[[lists]]$Group = "EEG"
  df.Pbs[[lists]]$Group[df.Pbs[[lists]]$PID %in% TMS.PIDs] = "TMS"
  table(df.Pbs[[lists]]$Group)
}
remove(TMS.PIDs)

# Quality Check -----
# Continuous block numbers
```

```

for (lists in names(df.Pbs)) {
}

qc = list()
# summarise trials
# Continuous block numbers
for (lists in names(df.Pbs)) {
  check_summary(df.Pbs[[lists]]$UP1DN2)
}
remove(powerband, powerbandname)

# Outliers -----

for (lists in names(df.Pbs)) {
  channel_list = grep(pattern = "^Ch\\d+", x = colnames(df.Pbs[[lists]]))
  channel_list = colnames(df.Pbs[[lists]])[channel_list]
  for (channel in channel_list) {
    Stability.Outliers = Check_For_Outliers(chunk.data.y = df.Pbs[[lists]][[channel]],
      pdf.path.name = paste0("./Processed Data/EEG
Power/Channel Outliers/",
                                lists, "-", channel, "-
Outliers.pdf"))
    df.Pbs[[lists]][[paste0(channel, ".Outliers")]] = Stability.Outliers$outliers
  }
}

remove(Stability.Outliers)
# Incorporate Achievers -----
load(file = "./Processed Data/CA Data/df-Achievers.R")
df.comp = list()
# nr 10 not in here at all and nr 67 "falls out", fixed both
for (lists in names(df.Pbs)) {
  df.Pbs[[lists]]$UID = interaction(df.Pbs[[lists]]$PID, df.Pbs[[lists]]$Ses,
df.Pbs[[lists]]$Block, sep = ".")
  df.comp[[lists]] = merge(df.Pbs[[lists]], df.achievers,
    by = intersect(names(df.Pbs[[lists]]),
names(df.achievers)),
    all.x = TRUE)
}

# merge with StabDist also?
load(file = "./Processed Data/StabDist/Processed/CA-StabDist-Achievers.R")
for (lists in names(df.comp)) {
  df.comp[[lists]] = merge(df.comp[[lists]], df.StabDist.complete,
    by = intersect(names(df.comp[[lists]]),
names(df.StabDist.complete)),
    all.x = TRUE)
}

# Block with only 9-20
df.eegbci = list()
for (lists in names(df.comp)) {
  df.eegbci[[lists]] = df.comp[[lists]][df.comp[[lists]]$Ses %in% c(3, 4, 5), ]
}

save(df.comp, file = "./Processed Data/EEG Power/Processed Data/df-Power-CA.R")
save(df.eegbci, file = "./Processed Data/EEG Power/Processed Data/df-Power-CA-eegbci.R")

```


10.4.10 Power 2 (R)

This script makes all the linear models reported.

```
# This script we will look at the Power Metrics

# Init -----
# clear all
rm(list = ls())

# Libraries
library(lmerTest)
library(lme4)
library(doParallel)
library(ggplot2)
library(extrafont)
library(ggpubr)
library(see)
library(readxl)
library(ggh4x)

# set wd
work_dir = ""
setwd(work_dir)

# get useful functions
source(file = "../../Code/R Code/R_useful_funcs_share.R")

# there are multiple files to look at here:
power.files = list.files(path = "../Processed Data/EEG Power/Data/", pattern = ".csv")

# read all of them in:
load(file = "../Processed Data/EEG Power/Processed Data/df-Power-CA-eegbci.R")
load(file = "../Processed Data/EEG Power/Processed Data/df-Power-CA.R")

# exclude where CA is NA, and PID 9:
for (lists in names(df.eegbci)) {
  df.eegbci[[lists]] = df.eegbci[[lists]][!is.na(df.eegbci[[lists]]$CA), ]
  df.eegbci[[lists]] = df.eegbci[[lists]][!df.eegbci[[lists]]$PID == 9, ]
}
for (lists in names(df.comp)) {
  df.comp[[lists]] = df.comp[[lists]][!is.na(df.comp[[lists]]$CA), ]
  df.comp[[lists]] = df.comp[[lists]][!df.comp[[lists]]$PID == 9, ]
}

# All in One DF -----
# Pick Cluster of Electrodes:
# read in electrode names:
elec.names = read.table('/ Channel Name List.txt');
# remove the weird values: (T4), (Inion), (T3)
elec.names = elec.names$V1[!(elec.names$V1 %in% c("(T4)", "(inion)", "(T3)"))]
# TMS Cluster
tms.clust.ind = which(elec.names %in% c("CP3", "C3", "CP1"))
eeg.clust.ind = which(elec.names %in% c("C1", "Cz", "FC1"))

# cycle through all lists
df.power = NULL
for (powerband in names(df.eegbci)) {

  # get current df
  current.df = df.eegbci[[powerband]]

  # get col names
```

```

channel.cols = grepl("Ch", colnames(current.df))

# get cluster col indexes (and outliers)
tms.cluster = colnames(current.df) %in% paste0("Ch", tms.clust.ind)
tms.cluster.outliers = colnames(current.df) %in% paste0("Ch", tms.clust.ind,
".Outliers")

# create df you're gonna add by removing all channels
df.to.add = current.df[,!channel.cols]
# if it's first run use it to create df
if (is.null(df.power)){
  df.power = df.to.add
}

# create second df with all channel info so you can manipulate names
df.channels = current.df[c(tms.cluster)]
df.channels$mean = rowMeans(df.channels)
df.channels$mean.outliers = ID.outliers(df.channels$mean)
df.channels = cbind(df.channels, current.df[,tms.cluster.outliers])
# Chang the names
colnames(df.channels) = paste0(powerband, "_", colnames(df.channels))
# add all the rest so merging is consistent
df.channels = cbind(df.to.add, df.channels)

# add df to results
df.power = merge(df.power, df.channels)

# clean up to ensure no dupes
remove(df.to.add, df.channels, tms.cluster, tms.cluster.outliers, channel.cols,
current.df)
}

# First Look at Models -----
Model_me_this(model.data = df.eegbci$`Global_1_&_45_Hz_Power`,
  model.formula = "CA ~ Continuous.Block * Group + (1|PID)",
  save_path = "../Processed Data/EEG Power/Models/Base Model/Base")
# Still works that's nice

# KSS? -----
df.power.kss = df.power[!is.na(df.power$KSS),]

## Individual Power Bands -----
models = list()
models$m0 = lmer(data = df.power, formula = CA ~ Group * Continuous.Block + (1|PID))
for (powerband in names(df.eegbci)) {
  save.path = paste0("../Processed Data/EEG Power/Models/Power Bands CA/", powerband, "-
FullInt/", powerband, "-FullInt")
  current_model = Model_me_this(model.data = df.power,
    model.formula = paste0("CA ~ Group * Continuous.Block *
\\", powerband, "_mean\\ + (1|PID)"),
    save_path = save.path)
  models[[powerband]] = current_model$LM$Model
}

anova(models$`Global_1_&_45_Hz_Power`, models$`HighAlpha_11_&_14_Hz_Power`,
models$`HighBeta_22_&_31_Hz_Power`,
  models$`HighDelta_1_&_5_Hz_Power`, models$`LowAlpha_8_&_11_Hz_Power`,
models$`LowBeta_14_&_22_Hz_Power`,
  models$`LowGamma_31_&_45_Hz_Power`, models$`OpenVibe_8_&_30_Hz_Power`,
models$`Theta_5_&_8_Hz_Power`)
anova(models$m0, models$`LowAlpha_8_&_11_Hz_Power`)
# # read in all models and anova them
# # there are multiple files to look at here:
## Individual Power Bands Hand or Cross instead of CA -----
-----

```

```

for (powerband in names(df.eegbci)) {
  save.path = paste0("./Processed Data/EEG Power/Models/Power Bands Hand/KSS/",
    powerband, "-FullInt/", powerband, "-FullInt")
  Model_me_this(model.data = df.power.kss,
    model.formula = paste0("\`Success Hand\` ~ KSS + Group * Continuous.Block *
  \`, powerband, "_mean\` + (1|PID)"),
    save_path = save.path)
}

for (powerband in names(df.eegbci)) {
  save.path = paste0("./Processed Data/EEG Power/Models/Power Bands Cross/KSS/",
    powerband, "-FullInt/", powerband, "-FullInt")
  Model_me_this(model.data = df.power.kss,
    model.formula = paste0("\`Success Cross\` ~ KSS + Group * Continuous.Block
  * \`, powerband, "_mean\` + (1|PID)"),
    save_path = save.path)
}

# All Powerbands -----
# have rank deficient matrix because of too many, also correlations global/openvibe
powerbands.interest = names(df.eegbci)[c(2:7, 9)]
# just alphabeta?
powerbands.alphabeta = names(df.eegbci)[grepl("Alpha", names(df.eegbci)) | grepl("Beta",
  names(df.eegbci))]

save.path = paste0("./Processed Data/EEG Power/Models/All Power Bands/Alpha-Beta/Alpha-
Beta")
powerbands.groupint = paste0("\`, powerbands.alphabeta, "_mean\`:Group")
powerbands.blockint = paste0("\`, powerbands.alphabeta, "_mean\`:Continuous.Block")
powerbands.blockgroupint = paste0("\`, powerbands.alphabeta,
  "_mean\`:Continuous.Block:Group")
powerbands = paste(powerbands.groupint, powerbands.blockint, powerbands.blockgroupint, sep
  = " + ", collapse = " + ")
Model_me_this(model.data = df.power,
  model.formula = paste0("CA ~ Group * Continuous.Block * ", powerbands, " +
  (1|PID)"),
  save_path = save.path)

# Look at Variables -----
# need electrode selection for this to work
Var.of.interest = c("PID", "Continuous.Block", "Group", "Distance UP:DN", "Distance
Rest:UP",
  "Distance Rest:DN", "Stability of Rest", "Stability of UP", "Stability
of DN", "KSS", "CA",
  "Success Cross", "Success Hand", "Failure Bias", "Sex", "Age", "KVIQ
Total", "Achiever.kss",
  "Achiever.m", "t.zero", "t.lastS", "t.Slope", "FirstVLast",
  "First.Distribution", "Second.Distribution")
require(psych)
for (list.name in names(df.comp)) {
  df.eegbci[[list]] = df.comp[[list]][df.comp[[list]]$Ses %in% c(3, 4, 5), ]
}
pdf(file = paste0("./Processed Data/EEG Power/Variables/Var-", list.name, ".pdf"), width =
50, height = 50)
pairs.panels(df.EEGBCI.SecondDist[df.EEGBCI.SecondDist$Dist.Outliers == FALSE, ],
  method = "pearson", # correlation method
  hist.col = "#00AFBB",
  density = TRUE, # show density plots
  ellipses = TRUE, # show correlation ellipses
  stars = TRUE
)
dev.off()
# Making some Plots -----

# Ideally as function but that's later first get the plot we want
# prepare variables

```

```

plot.data = df.power
achievers.sign = c('')
plot.data$achievers = plot.data$PID %in% achievers.sign
plot.data$achievers = ifelse(plot.data$achievers, "Control", "No Control")
save.path = "../Processed Data/EEG Power/Power Plots/"
df.power$Session = as.factor(df.power$Session)

power_plot <- function(plot.data, save.path, powerband){
  # browser()
  # get just powerbandname for saving
  powerband.name = strsplit(powerband, "_")[[1]][2]
  powerband.type = strsplit(powerband, "-")[[1]][1]
  # add bandname folder
  save.path = paste0(save.path, powerband.type, "-", powerband.name)
  # add in space
  powerband.name = gsub("High", "High ", powerband.name)
  powerband.name = gsub("Low", "Low ", powerband.name)

  # Check if path does not exist
  if (!exists((save.path))){
    # create path then
    dir.create((save.path))
  }

  # get outliers:
  plot.data$outliers = Check_For_Outliers(chunk.data.y = df.power[[powerband]],
                                           pdf.path.name = paste0(save.path, "/outliers-",
powerband.name, ".pdf"))
  # exclude outliers
  plot.data = plot.data[!plot.data$outliers$outliers, ]

  # make session a factor
  plot.data$Session = factor(plot.data$Session)

  # title of ggplot
  title.ggplot = paste0("Association Between ",
                        gsub(" mean", "", gsub("& ", "-", gsub("_", " ", powerband))),
                        " And CA Split by Session, Control and Group")
  title.ggplot = gsub("High", "High ", title.ggplot)
  title.ggplot = gsub("Low", "Low ", title.ggplot)
  title.ggplot = gsub("relpower", "Relative Power,", title.ggplot)
  title.ggplot = gsub("percchange", "Percentage Change,", title.ggplot)
  metric.type = ifelse(grepl("Percentage Change", title.ggplot), "Percentage Change",
"Relative Power")

  # do the plot
  actual.plot = ggplot(data = plot.data, aes(x = CA, y = plot.data[[powerband]], col =
Session)) +
    facet_nested_wrap( ~ achievers + Group, nrow = 2, nest_line = TRUE) +
    geom_point(alpha = 0.5) +
    theme_minimal() +
    theme(text = element_text(family = "Times"),
          legend.position = "bottom",
          panel.spacing = unit(0.3, "lines"),
          panel.border = element_rect(fill = NA, linetype = 1, colour = "lightgray"),
          strip.background = element_rect(fill = NA, linetype = 1, colour = "lightgray",
linewidth = 0.5)) +
    geom_smooth(method = "lm", se = FALSE) +
    # geom_line(data = cbind(model.data, pred = predict(lm.model)), aes(y = pred), size =
1) +
    labs(x = "Classification Accuracy [%]", y = paste0(metric.type, ", ", powerband.name, "
TMS Cluster (CP3, C3, CP1)"),
         title = title.ggplot) +
    scale_colour_viridis_d(alpha = 0.8, direction = 1) +
    scale_fill_viridis_d(alpha = 0.8, direction = 1)
  actual.plot

  ggsave(filename = paste0(save.path, "/", powerband.name, ".pdf"), width = 9, height = 7)

```

```
}  
  
for (powerband in names(df.eegbci)) {  
  power_plot(plot.data = plot.data,  
             save.path = "./Processed Data/EEG Power/Power Plots/",  
             powerband = paste0(powerband, "_mean"))  
}
```

10.4.11 Power 3 (R)

This script makes a lot of the power-related plots.

```
library(ggh4x)
library(patchwork)
library(scales)

# First Look at Models -----

Model_me_this(model.data = df.eegbci`percchange_Global_1_&_45_Hz_Power`,
               model.formula = "CA ~ Continuous.Block * Group + (1|PID)",
               save_path = "./Processed Data/EEG Power/Models/Base Model/Base")
# Still works that's nice

# KSS? -----
df.power.kss = df.power[!is.na(df.power$KSS),]

## Individual Power Bands -----
models = list()
models$m0 = lmer(data = df.power, formula = CA ~ Group * Continuous.Block + (1|PID))
for (powerband in names(df.eegbci)) {
  save.path = paste0("./Processed Data/EEG Power/Models/Power Bands CA/", powerband, "-FullInt/", powerband, "-FullInt")
  current_model = Model_me_this(model.data = df.power,
                                model.formula = paste0("CA ~ Group * Continuous.Block *`", powerband, "_mean` + (1|PID)"),
                                save_path = save.path)
  models[[powerband]] = current_model$LM$Model
}

anova(models$`percchange_Global_1_&_45_Hz_Power`,
models$`percchange_HighAlpha_11_&_14_Hz_Power`,
models$`percchange_HighBeta_22_&_31_Hz_Power`,
models$`percchange_HighDelta_1_&_5_Hz_Power`,
models$`percchange_LowAlpha_8_&_11_Hz_Power`, models$`percchange_LowBeta_14_&_22_Hz_Power`,
models$`percchange_LowGamma_31_&_45_Hz_Power`,
models$`percchange_OpenVibe_8_&_30_Hz_Power`, models$`percchange_Theta_5_&_8_Hz_Power`)
anova(models$m0, models$`percchange_LowAlpha_8_&_11_Hz_Power`)

anova(models$`rempower_Global_1_&_45_Hz_Power`,
models$`rempower_HighAlpha_11_&_14_Hz_Power`, models$`rempower_HighBeta_22_&_31_Hz_Power`,
models$`rempower_HighDelta_1_&_5_Hz_Power`,
models$`rempower_LowAlpha_8_&_11_Hz_Power`, models$`rempower_LowBeta_14_&_22_Hz_Power`,
models$`rempower_LowGamma_31_&_45_Hz_Power`,
models$`rempower_OpenVibe_8_&_30_Hz_Power`, models$`rempower_Theta_5_&_8_Hz_Power`)

for (powerband in names(df.eegbci)) {
  save.path = paste0("./Processed Data/EEG Power/Models/Power Bands CA/KSS/", powerband, "-FullInt/", powerband, "-FullInt")
  current_model = Model_me_this(model.data = df.power.kss,
                                model.formula = paste0("CA ~ KSS + Group * Continuous.Block *`", powerband, "_mean` + (1|PID)"),
                                save_path = save.path)
  models[[powerband]] = current_model$LM$Model
}

## Individual Power Bands Hand or Cross instead of CA -----
-----
for (powerband in names(df.eegbci)) {
  save.path = paste0("./Processed Data/EEG Power/Models/Power Bands Hand/KSS/", powerband, "-FullInt/", powerband, "-FullInt")
  Model_me_this(model.data = df.power.kss,
```

```

        model.formula = paste0("\`Success Hand\` ~ KSS + Group * Continuous.Block *
\`", powerband, "_mean\` + (1|PID)"),
        save_path = save.path)
    }

for (powerband in names(df.eegbci)) {
    save.path = paste0("./Processed Data/EEG Power/Models/Power Bands Cross/KSS/",
powerband, "-FullInt/", powerband, "-FullInt")
    Model_me_this(model.data = df.power.kss,
        model.formula = paste0("\`Success Cross\` ~ KSS + Group * Continuous.Block
* \`", powerband, "_mean\` + (1|PID)"),
        save_path = save.path)
    }

for (powerband in names(df.eegbci)) {
    save.path = paste0("./Processed Data/EEG Power/Models/Power Bands Hand/", powerband, "-
FullInt/", powerband, "-FullInt")
    Model_me_this(model.data = df.power.kss,
        model.formula = paste0("\`Success Hand\` ~ Group * Continuous.Block * \`",
powerband, "_mean\` + (1|PID)"),
        save_path = save.path)
    }

for (powerband in names(df.eegbci)) {
    save.path = paste0("./Processed Data/EEG Power/Models/Power Bands Cross/", powerband, "-
FullInt/", powerband, "-FullInt")
    Model_me_this(model.data = df.power.kss,
        model.formula = paste0("\`Success Cross\` ~ Group * Continuous.Block * \`",
powerband, "_mean\` + (1|PID)"),
        save_path = save.path)
    }

# All Powerbands -----
# have rank deficient matrix because of too many, also correlations global/openvibe
powerbands.interest = names(df.eegbci)[c(2:7, 9)]
# just alphabeta?
powerbands.alphabeta = names(df.eegbci)[grepl("Alpha", names(df.eegbci)) | grepl("Beta",
names(df.eegbci))]
powerbands.list = list()
powerbands.list$percchange = powerbands.alphabeta[grepl("percchange",
powerbands.alphabeta)]
powerbands.list$relpow = powerbands.alphabeta[!grepl("percchange", powerbands.alphabeta)]

for (powerband.listname in names(powerbands.list)) {
    save.path = paste0("./Processed Data/EEG Power/Models/All Power Bands/Alpha-Beta/",
powerband.listname, '/', powerband.listname)
    powerbands.groupint = paste0("\`", powerbands.list[[powerband.listname]],
"_mean\`:Group")
    powerbands.blockint = paste0("\`", powerbands.list[[powerband.listname]],
"_mean\`:Continuous.Block")
    powerbands.blockgroupint = paste0("\`", powerbands.list[[powerband.listname]],
"_mean\`:Continuous.Block:Group")
    powerbands = paste(powerbands.groupint, powerbands.blockint, powerbands.blockgroupint,
sep = " + ", collapse = " + ")

    Model_me_this(model.data = df.power,
        model.formula = paste0("CA ~ Group * Continuous.Block * ", powerbands, " +
(1|PID)"),
        save_path = save.path)
    }

# Look at Variables -----
# need electrode selection for this to work
Var.of.interest = c("PID", "Continuous.Block", "Group", "Distance UP:DN", "Distance
Rest:UP",

```

```

        "Distance Rest:DN", "Stability of Rest", "Stability of UP", "Stability
of DN", "KSS", "CA",
        "Success Cross", "Success Hand", "Failure Bias", "Sex", "Age", "KVIQ
Total", "Achiever.kss",
        "Achiever.m", "t.zero", "t.lastS", "t.Slope", "FirstVLast",
"First.Distribution", "Second.Distribution")
require(psych)
for (list.name in names(df.comp)) {
  df.eegbci[[lists]] = df.comp[[lists]][df.comp[[lists]]$Ses %in% c(3, 4, 5), ]
}
pdf(file = paste0("./Processed Data/EEG Power/Variables/Var-", list.name, ".pdf"), width =
50, height = 50)
pairs.panels(df.EEGBCI.SecondDist[df.EEGBCI.SecondDist$Dist.Outliers == FALSE , ],
  method = "pearson", # correlation method
  hist.col = "#00AFBB",
  density = TRUE, # show density plots
  ellipses = TRUE, # show correlation ellipses
  stars = TRUE
)
dev.off()
# Making some Plots -----

# Ideally as function but that's later first get the plot we want
# prepare variables
plot.data = df.power
achievers.sign = c()
plot.data$achievers = plot.data$PID %in% achievers.sign
plot.data$achievers = ifelse(plot.data$achievers, "Control", "No Control")
save.path = "./Processed Data/EEG Power/Power Plots/"
df.power$Session = as.factor(df.power$Session)

power_plot <- function(plot.data, save.path, powerband){
  # browser()
  # get just powerbandname for saving
  powerband.type = strsplit(powerband, "_")[[1]][1]
  powerband.name = strsplit(powerband, "_")[[1]][2]
  # add bandname folder
  save.path = paste0(save.path, powerband.type, powerband.name)
  # add in space
  powerband.name = gsub("High", "High ", powerband.name)
  powerband.name = gsub("Low", "Low ", powerband.name)

  # Check if path does not exist
  if (!exists((save.path))){
    # create path then
    dir.create((save.path))
  }

  # get outliers:
  plot.data.outliers = Check_For_Outliers(chunk.data.y = df.power[[powerband]],
                                          pdf.path.name = paste0(save.path, "/outliers-",
powerband.name, ".pdf"))
  # exclude outliers
  plot.data = plot.data[!plot.data.outliers$outliers, ]

  # make session a factor
  plot.data$Session = factor(plot.data$Session)

  # title of ggplot
  title.ggplot = paste0("Correlation Between ",
                        gsub(" mean", "", gsub("& ", "-", gsub("_", " ", powerband))),
                        " And Classification Accuracy Split by Session")
  title.ggplot = gsub("High", "High ", title.ggplot)
  title.ggplot = gsub("Low", "Low ", title.ggplot)

  # do the plot
  actual.plot = ggplot(data = plot.data, aes(x = CA, y = plot.data[[powerband]], col =
Session)) +

```



```

facet_nested_wrap( ~ achievers + Group, nrow = 2, nest_line = TRUE) +
geom_point(alpha = 0.5) +
theme_minimal() +
theme(text = element_text(family = "Times"),
      legend.position = "bottom",
      panel.spacing = unit(0.3, "lines"),
      panel.border = element_rect(fill = NA, linetype = 1, colour = "lightgray"),
      strip.background = element_rect(fill = NA, linetype = 1, colour = "lightgray",
linewidth = 0.5)) +
geom_smooth(method = "lm", se = FALSE) +
# geom_line(data = cbind(model.data, pred = predict(lm.model)), aes(y = pred), size =
1) +
labs(x = "Classification Accuracy [%]", y = paste0(powerband.name, " Mean Power, TMS
Cluster (CP3, C3, CP1)"),
      title = title.ggplot) +
scale_colour_viridis_d(alpha = 0.8, direction = 1) +
scale_fill_viridis_d(alpha = 0.8, direction = 1)
actual.plot

ggsave(filename = paste0(save.path, "/", powerband.name, ".pdf"), width = 9, height = 7)
}

for (powerband in names(df.eegbci)) {
  power_plot(plot.data = plot.data,
            save.path = "./Processed Data/EEG Power/Power Plots/",
            powerband = paste0(powerband, "_mean"))
}

# Dotplot Function-----
Nice_Dotplot <- function(plot.data, name.powerband, save.path){
  # browser()
  ## Initialise things -----

  ggplot.data = plot.data
  # set naming
  metric.type = ifelse(grepl("percchange", name.powerband), TRUE, FALSE)
  name.title.type = ifelse(metric.type, "Percentage Change", "Relative Power")
  name.ylab = gsub(ifelse(metric.type, "percchange ", "relpower "),
                  ifelse(metric.type, "Percentage Change ", "Relative Power "),
                  gsub("mean", "Mean",
                      gsub(ifelse(grepl("High", name.powerband), "High", "Low"),
                          ifelse(grepl("High", name.powerband), "High ", "Low "),
                          gsub(" Hz", "Hz",
                              gsub(" & ", "-",
                                  gsub("-", " ", name.powerband))))))

  gg.plot.data = data.frame()
  # eliminate outliers by creating a new datastructure and only adding non outliers
  for (sessions in unique(ggplot.data$Session)) {
    # separate data
    current.data.up = ggplot.data[ggplot.data$Session == sessions & ggplot.data$UP1DN2 ==
1,]
    current.data.down = ggplot.data[ggplot.data$Session == sessions & ggplot.data$UP1DN2 ==
2,]

    # take out outliers
    current.data.up.nout =
current.data.up[!ID.outliers(current.data.up[[name.powerband]]),]
    current.data.down.nout =
current.data.down[!ID.outliers(current.data.down[[name.powerband]]),]

    # Link Data back together
    gg.plot.data = rbind(gg.plot.data, current.data.up.nout)
    gg.plot.data = rbind(gg.plot.data, current.data.down.nout)
  }
}

```

```

## Test data -----

# T.test data between sessions, within participants
necess.tests = unique(interaction(gg.plot.data$Group, gg.plot.data$UP1DN2))
paired.tests = list()
for (group.condition in necess.tests) {
  # prepare for pairing
  data.test = gg.plot.data
  group = ifelse(grepl("EEG", group.condition), "EEG", "TMS")
  condition = ifelse(grepl("1", group.condition), 1, 2)

  # prepare merge indices
  data.test$merge.UID = interaction(data.test$PID, data.test$Block)
  ind.s3 = c(data.test$Sess == 3 & data.test$Group == group & data.test$UP1DN2 ==
condition)
  ind.s5 = c(data.test$Sess == 5 & data.test$Group == group & data.test$UP1DN2 ==
condition)
  data.s3 = data.frame(cbind(data.test[[name.powerband]][ind.s3],
as.character(data.test$merge.UID[ind.s3])))
  data.s5 = data.frame(cbind(data.test[[name.powerband]][ind.s5],
as.character(data.test$merge.UID[ind.s5])))

  # Make paired
  paired.data = MakePaired(data.s3, data.s5, "X2")
  colnames(paired.data$result) = c("UID", "Sess3", "Sess5")

  # Make test
  paired.tests[[group.condition]] = PairedTest(x = as.numeric(paired.data$result$Sess3),
y = as.numeric(paired.data$result$Sess5),
testName = paste0(group.condition, " Sess
3 VS Sess 5"))
}

# T.test data between Conditions, within participants
necess.tests = unique(interaction(c(3,5), gg.plot.data$Group))
paired.tests.upvdn = list()
for (session.group in necess.tests) {
  # prepare for pairing
  data.test = gg.plot.data
  group = ifelse(grepl("EEG", session.group), "EEG", "TMS")
  sess = ifelse(grepl("3", session.group), 3, 5)

  # prepare merge indices
  data.test$merge.UID = interaction(data.test$PID, data.test$Block)
  ind.up = c(data.test$UP1DN2 == 1 & data.test$Group == group & data.test$Session ==
sess)
  ind.down = c(data.test$UP1DN2 == 2 & data.test$Group == group & data.test$Session ==
sess)
  data.up = data.frame(cbind(data.test[[name.powerband]][ind.up],
as.character(data.test$merge.UID[ind.up])))
  data.down = data.frame(cbind(data.test[[name.powerband]][ind.down],
as.character(data.test$merge.UID[ind.down])))

  # Make paired
  paired.data = MakePaired(data.up, data.down, "X2")
  colnames(paired.data$result) = c("UID", "Up", "Down")

  # Make test
  paired.tests.upvdn[[session.group]] = PairedTest(x = as.numeric(paired.data$result$Up),
y =
as.numeric(paired.data$result$Down),
testName = paste0(session.group, " Up
VS Down"))
}

# T.test data between participants, same session
necess.tests = unique(interaction(c(3, 5), gg.plot.data$UP1DN2))
ind.t.tests = list()

```

```

for (Sess.condition in necess.tests) {
  # prepare for pairing
  data.test = gg.plot.data
  sess = ifelse(grepl("3", Sess.condition), 3, 5)
  condition = ifelse(grepl("1", Sess.condition), 1, 2)

  # prepare merge indices
  ind.EEG = c(data.test$Group == "EEG" & data.test$Session == sess & data.test$UP1DN2 ==
condition)
  ind.TMS = data.test$Group == "TMS" & data.test$Session == sess & data.test$UP1DN2 ==
condition
  data.EEG = data.test[[name.powerband]][ind.EEG]
  data.TMS = data.test[[name.powerband]][ind.TMS]

  # Make test
  ind.t.tests[[Sess.condition]] = IndependentTest(x = data.EEG,
                                                    y = data.TMS,
                                                    testName = paste0(Sess.condition, " EEG
VS TMS"))
}

# Plot the plot -----
# Plot plots
# get the max x value for brackets
y.max = max(gg.plot.data[[name.powerband]])
y.min = min(gg.plot.data[[name.powerband]])
if (name.title.type == "Percentage Change"){
  bracket.tip.length = y.max/20000 #10000
  bracket.step.inc = y.max/10 #5
} else {
  bracket.tip.length = y.max/500
  bracket.step.inc = y.max/30
}
bracket.vjust = 3 #2.5
bracket.label.size = 2.5

gg.plot = ggplot(data = gg.plot.data, aes(x = interaction(Sess, Group, UP1DN2),
                                                    y = gg.plot.data[[name.powerband]],
                                                    color = Group)) +
  # geom_violinhalf(position = position_nudge(x = -.075, y = 0), flip = TRUE) +
  # facet_wrap(~ achievers, nrow = 2) +
  geom_jitter(width = 0.04) +
  geom_boxplot(width = 0.1, position = position_nudge(x = 0.15, y = 0),
              outlier.alpha = 0) +
  theme_light() +
  xlab(label = "Session Number, Group") +
  scale_x_discrete(limits = c("3.EEG.1", "3.EEG.2", "3.TMS.1", "3.TMS.2",
                              "5.EEG.1", "5.EEG.2", "5.TMS.1", "5.TMS.2"),
                  labels = c("Session 3, EEG, Up", "Session 3, EEG, Down", "Session 3,
TMS, Up", "Session 3, TMS, Down",
                              "Session 5, EEG, Up", "Session 5, EEG, Down", "Session 5,
TMS, Up", "Session 5, TMS, Down")) +
  ylab(label = name.ylab) +
  labs(title = paste0(name.title.type, ", TMS vs EEG Group, Comparison of Third Session
vs Last Session")) +
  theme(text = element_text(family = "Times"),
        axis.text.x = element_text(angle = 35, hjust = 1)) +
  scale_color_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_fill_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +

  # add brackets for paired tests
  geom_bracket(inherit.aes = FALSE,
              xmin = "3.TMS.1", xmax = "5.TMS.1", y.position = y.max +
3*bracket.step.inc , tip.length = bracket.tip.length,
              label = paste(signif(paired.tests$TMS.1$test$p.value, 2)),
              family = "Times", label.size = bracket.label.size) +
  geom_bracket(inherit.aes = FALSE,

```

```

      xmin = "3.TMS.2", xmax = "5.TMS.2", y.position = y.max +
4*bracket.step.inc , tip.length = bracket.tip.length,
      label = paste(signif(paired.tests$TMS.2$test$p.value, 2)),
      family = "Times", label.size = bracket.label.size) +
  geom_bracket(inherit.aes = FALSE,
    xmin = "3.EEG.1", xmax = "5.EEG.1", y.position = y.max +
1*bracket.step.inc , tip.length = bracket.tip.length,
      label = paste(signif(paired.tests$EEG.1$test$p.value, 2)),
      family = "Times", label.size = bracket.label.size) +
  geom_bracket(inherit.aes = FALSE,
    xmin = "3.EEG.2", xmax = "5.EEG.2", y.position = y.max +
2*bracket.step.inc, tip.length = bracket.tip.length,
      label = paste(signif(paired.tests$EEG.2$test$p.value, 2)),
      family = "Times", label.size = bracket.label.size) +
  # let's add brackets for between condition comparison
  geom_bracket(inherit.aes = FALSE,
    xmin = "3.EEG.1", xmax = "3.EEG.2", y.position = y.min -
1*bracket.step.inc , tip.length = -bracket.tip.length, vjust = bracket.vjust,
      label = paste(signif(paired.tests.upvdm$`3.EEG`$test$p.value, 2)),
      family = "Times", label.size = bracket.label.size) +
  geom_bracket(inherit.aes = FALSE,
    xmin = "3.TMS.1", xmax = "3.TMS.2", y.position = y.min -
1*bracket.step.inc , tip.length = -bracket.tip.length, vjust = bracket.vjust,
      label = paste(signif(paired.tests.upvdm$`3.TMS`$test$p.value, 2)),
      family = "Times", label.size = bracket.label.size) +
  geom_bracket(inherit.aes = FALSE,
    xmin = "5.EEG.1", xmax = "5.EEG.2", y.position = y.min -
1*bracket.step.inc , tip.length = -bracket.tip.length, vjust = bracket.vjust,
      label = paste(signif(paired.tests.upvdm$`5.EEG`$test$p.value, 2)),
      family = "Times", label.size = bracket.label.size) +
  geom_bracket(inherit.aes = FALSE,
    xmin = "5.TMS.1", xmax = "5.TMS.2", y.position = y.min -
1*bracket.step.inc , tip.length = -bracket.tip.length, vjust = bracket.vjust,
      label = paste(signif(paired.tests.upvdm$`5.TMS`$test$p.value, 2)),
      family = "Times", label.size = bracket.label.size) +

  # Brackets for the between group comparison
  geom_bracket(inherit.aes = FALSE,
    xmin = "3.EEG.1", xmax = "3.TMS.1", y.position = y.min -
2*bracket.step.inc , tip.length = -bracket.tip.length, vjust = bracket.vjust,
      label = paste(signif(ind.t.tests$`3.1`$test$p.value, 2)),
      family = "Times", label.size = bracket.label.size) +
  geom_bracket(inherit.aes = FALSE,
    xmin = "3.EEG.2", xmax = "3.TMS.2", y.position = y.min -
3*bracket.step.inc , tip.length = -bracket.tip.length, vjust = bracket.vjust,
      label = paste(signif(ind.t.tests$`3.2`$test$p.value, 2)),
      family = "Times", label.size = bracket.label.size) +
  geom_bracket(inherit.aes = FALSE,
    xmin = "5.EEG.1", xmax = "5.TMS.1", y.position = y.min -
2*bracket.step.inc , tip.length = -bracket.tip.length, vjust = bracket.vjust,
      label = paste(signif(ind.t.tests$`5.1`$test$p.value, 2)),
      family = "Times", label.size = bracket.label.size) +
  geom_bracket(inherit.aes = FALSE,
    xmin = "5.EEG.2", xmax = "5.TMS.2", y.position = y.min -
3*bracket.step.inc , tip.length = -bracket.tip.length, vjust = bracket.vjust,
      label = paste(signif(ind.t.tests$`5.2`$test$p.value, 2)),
      family = "Times", label.size = bracket.label.size)

# Save everything -----
# check if folder exists, and if not create it
name.file = paste0(save.path, name.title.type, " ", name.ylab, "/", name.title.type, " ",
name.ylab)
# Check if path does not exist
if (!exists(dirname(name.file))) {
  # create path then
  dir.create(dirname(name.file))
}

```

```

# Save Plot
ggsave(filename = paste0(name.file, ".pdf"), width = 8, height = 8)

# Save tests
write(capture.output(paired.tests), file = paste0(name.file, "between groups.txt"))
write(capture.output(ind.t.tests), file = paste0(name.file, "between sessions.txt"))
write(capture.output(paired.tests.upvbn), file = paste0(name.file, "between
conditions.txt"))
return(gg.plot)
}

# Do all the work -----

# get all plots that needs plotting
plots.to.do = names(plot.data[grepl("_mean", names(plot.data))])
plots.to.do = plots.to.do[!grepl("outliers", plots.to.do)]

for (name.powerband in names(plot.data[plots.to.do])) {
  Nice_Dotplot(plot.data = plot.data,
               name.powerband = name.powerband,
               save.path = "../Processed Data/EEG Power/Power Plots/Dotplots/")
}

# try half plots?
ach.plot = list()
for (name.powerband in names(plot.data[plots.to.do])) {
  ach.plot[[name.powerband]] = Nice_Dotplot(plot.data = plot.data[plot.data$achievers ==
"Control", ],
                                             name.powerband = name.powerband,
                                             save.path = "../Processed Data/EEG Power/Power
Plots/Dotplots/Cont_")
}

# try half plots?
nach.plot = list()
for (name.powerband in names(plot.data[plots.to.do])) {
  nach.plot[[name.powerband]] = Nice_Dotplot(plot.data = plot.data[plot.data$achievers ==
"No Control", ],
                                             name.powerband = name.powerband,
                                             save.path = "../Processed Data/EEG Power/Power
Plots/Dotplots/Ncont_")
}

# arrange the plots
comb.plot = list()
for (name.powerband in names(ach.plot)) {
  # find out scales
  ylims.ach = layer_scales(ach.plot[[name.powerband]])$y$get_limits()
  ylims.nach = layer_scales(nach.plot[[name.powerband]])$y$get_limits()
  ylim.min = min(ylims.ach, ylims.nach)
  ylim.max = max(ylims.ach, ylims.nach)

  # get plot title
  plot.title = ach.plot[[name.powerband]]$labels$title

  # modify first plot
  first.plot = ach.plot[[name.powerband]]
  first.plot = first.plot +
    labs(title = NULL, subtitle = "Controllers") +
    theme(legend.position = "none") +
    xlab(label = waiver())

  second.plot = nach.plot[[name.powerband]]
  second.plot = second.plot +
    labs(title = NULL, subtitle = "Noncontrollers") +
    ylab(label = NULL) +
    xlab(label = waiver())
}

```

```

combined.plot = first.plot +
  second.plot &
  scale_y_continuous(limits = c(ylim.min, ylim.max))

combined.plot = (combined.plot + plot_annotation(title = plot.title)) &
  theme(text = element_text('Times'))
comb.plot[[name.powerband]] = combined.plot

# Save Plot
save.path = "./Processed Data/EEG Power/Power Plots/Dotplots/"
name.file = paste0(save.path, 'facetplot_', name.powerband)
ggsave(filename = paste0(name.file, ".pdf"), plot = combined.plot, width = 9, height = 6)
}

```

10.4.12 CSP Processing (R)

This script processes the CSP files and proceeds to test the result.

```
# This script we will look at the Power Metrics

# Init -----
# clear all
rm(list = ls())

# Libraries
library(lmerTest)
library(lme4)
library(doParallel)
library(ggplot2)
library(extrafont)
library(ggpubr)
library(see)
library(readxl)
library(readr)
library(xml2)

# set wd
work_dir = ""
setwd(work_dir)

# get useful functions
source(file = "")

# get all nfs:
all.nfs = list.files("../", pattern = "^nf\\d+")
# get all files
all.files = character()
for (nf in all.nfs) {
  new.files = list.files(paste0("../", nf, "/EEG BCI/CSP/"), pattern = ".*-csp-.*\\.cfg")
  start_ind = (length(all.files) + 1)
  end_ind = length(all.files) + (length(new.files))
  all.files[start_ind:end_ind] = new.files
}

# get all pids
all.pids = character()
for (csp.files in all.files) {
  all.pids[length(all.pids)+1] = strsplit(csp.files, split = "-")[[1]][1]
}
all.pids = sort(unique(all.pids))

# Analysis -----
# form mask
csp.matrices = list()
pid.mean.csp = list()
pid.mean.csp.ztrans = list()
for (pid in all.pids) {
  current.files = all.files[grepl(paste0(pid, "-"), all.files)]

  # read in .cfg file
  csp.matrices[[pid]] = list()
  csp.file.path = paste0("../nf", parse_number(pid), "/EEG BCI/CSP/")
  for (config in current.files) {
    current.csp.file.path = paste0(csp.file.path, config)
    curr.cfg = readLines(current.csp.file.path)
    curr.cfg = curr.cfg[2]
    curr.cfg = strsplit(curr.cfg, split = "</SettingValue>")[[1]]
    curr.cfg = strsplit(curr.cfg, split = "<SettingValue>")[[1]][2]
    curr.cfg = as.numeric(strsplit(curr.cfg, split = " ")[1])
  }
}
```

```

if (length(curr.cfg) != 1024){
  curr.cfg = matrix(curr.cfg, nrow = 16, ncol = 65, byrow = TRUE)
  curr.colsums.trans = colSums(curr.cfg)
  curr.colsums.trans = abs(curr.colsums.trans) - mean(curr.colsums.trans)
  curr.colsums.trans = data.frame(cbind(curr.colsums.trans,
ID.outliers(curr.colsums.trans, 3)))
  # and delete the last electrode with little to no activity
  curr.cfg = curr.cfg[,1:64]
  print(config)
  print(curr.colsums.trans)
} else {

  curr.cfg = matrix(curr.cfg, nrow = 16, ncol = 64, byrow = TRUE)
}
csp.matrices[[pid]][[config]] = curr.cfg
# save individ matrix
file.name = strsplit(config, split = ".cfg")[[1]][1]
write.csv(curr.cfg, file = paste0("./Processed Data/EEG Power/Processed Data/Session
CSP/", file.name, ".csv"))
}
# create PID average CSP
pid.averaged.csp = Reduce('+', csp.matrices[[pid]])/length(csp.matrices[[pid]])
pid.mean.csp[[pid]] = pid.averaged.csp
# z-transformed version?
pid.mean.csp.ztrans[[pid]] = z.transform(pid.averaged.csp)
write.csv(pid.averaged.csp, file = paste0("./Processed Data/EEG Power/Processed Data/PID
Averaged CSP/", pid, "-csp-mean.csv"))
}

# Averaged for Achievers and Group -----
# define groups
achievers.sign = c('')
achievers.trend = c('')
TMS = c('')
pid.numbers = parse_number(names(csp.matrices))

# define indices
ach.sign.ind = pid.numbers %in% achievers.sign
nach.sign.ind = !(pid.numbers %in% achievers.sign)
ach.trend.ind = pid.numbers %in% achievers.trend
tms.ind = pid.numbers %in% TMS

# combinatory indices
ach.sign.tms.ind = ach.sign.ind & tms.ind
nach.sign.tms.ind = !ach.sign.ind & tms.ind
ach.sign.eeg.ind = ach.sign.ind & !tms.ind
nach.sign.eeg.ind = !ach.sign.ind & !tms.ind

ach.trend.tms.ind = ach.trend.ind & tms.ind
nach.trend.tms.ind = !ach.trend.ind & tms.ind
ach.trend.eeg.ind = ach.trend.ind & !tms.ind
nach.trend.eeg.ind = !ach.trend.ind & !tms.ind

# create csp matrices:
csp.aggr = list()

csp.aggr$ach.sign = Reduce('+',
pid.mean.csp[ach.sign.ind])/length(pid.mean.csp[ach.sign.ind])
csp.aggr$Nach.sign = Reduce('+',
pid.mean.csp[!ach.sign.ind])/length(pid.mean.csp[!ach.sign.ind])

csp.aggr$ach.trend = Reduce('+',
pid.mean.csp[ach.trend.ind])/length(pid.mean.csp[ach.trend.ind])
csp.aggr$Nach.trend = Reduce('+',
pid.mean.csp[!ach.trend.ind])/length(pid.mean.csp[!ach.trend.ind])

csp.aggr$tms = Reduce('+', pid.mean.csp[tms.ind])/length(pid.mean.csp[tms.ind])
csp.aggr$eeg = Reduce('+', pid.mean.csp[!tms.ind])/length(pid.mean.csp[!tms.ind])

```



```

csp.aggr$ach.sign.tms = Reduce('+',
pid.mean.csp[ach.sign.tms.ind])/length(pid.mean.csp[ach.sign.tms.ind])
csp.aggr$nach.sign.tms = Reduce('+',
pid.mean.csp[nach.sign.tms.ind])/length(pid.mean.csp[nach.sign.tms.ind])
csp.aggr$ach.sign.eeg = Reduce('+',
pid.mean.csp[ach.sign.eeg.ind])/length(pid.mean.csp[ach.sign.eeg.ind])
csp.aggr$nach.sign.eeg = Reduce('+',
pid.mean.csp[nach.sign.eeg.ind])/length(pid.mean.csp[nach.sign.eeg.ind])

csp.aggr$ach.trend.tms = Reduce('+',
pid.mean.csp[ach.trend.tms.ind])/length(pid.mean.csp[ach.trend.tms.ind])
csp.aggr$nach.trend.tms = Reduce('+',
pid.mean.csp[nach.trend.tms.ind])/length(pid.mean.csp[nach.trend.tms.ind])
csp.aggr$ach.trend.eeg = Reduce('+',
pid.mean.csp[ach.trend.eeg.ind])/length(pid.mean.csp[ach.trend.eeg.ind])
csp.aggr$nach.trend.eeg = Reduce('+',
pid.mean.csp[nach.trend.eeg.ind])/length(pid.mean.csp[nach.trend.eeg.ind])

# Save all matrices
for (csp.mat in names(csp.aggr)) {
  csp.mat.name = gsub("\\.", "-", csp.mat)
  write.csv(csp.aggr[[csp.mat]], file = paste0("./Processed Data/EEG Power/Processed
Data/Categories/", csp.mat.name, ".csv"))
}

# Clusters -----
# TMS Cluster = CP3, C3, CP1
# EEG Cluster = C1, Cz, FC1

# read in electrode names:
elec.names = read.table('/ /Channel Name List.txt')
# remove the weird values: (T4), (Inion), (T3)
elec.names = elec.names$V1[!(elec.names$V1 %in% c("(T4)", "(inion)", "(T3)"))]
# TMS Cluster
tms.clust.ind = which(elec.names %in% c("CP3", "C3", "CP1"))
eeg.clust.ind = which(elec.names %in% c("C1", "Cz", "FC1"))

# TMS Clust Achievers -----
# creating lists
tms.ach.pids = names(csp.matrices[ach.sign.tms.ind])
tms.ach.weights = NULL
for (pid in tms.ach.pids) {
  new.weights = pid.mean.csp[[pid]][tms.clust.ind]
  start.ind = length(tms.ach.weights) + 1
  end.ind = start.ind + length(new.weights) - 1
  tms.ach.weights[start.ind:end.ind] = new.weights
}

# creating lists
eeg.ach.pids = names(csp.matrices[ach.sign.eeg.ind])
eeg.ach.weights = NULL
for (pid in eeg.ach.pids) {
  new.weights = pid.mean.csp[[pid]][tms.clust.ind]
  start.ind = length(eeg.ach.weights) + 1
  end.ind = start.ind + length(new.weights) - 1
  eeg.ach.weights[start.ind:end.ind] = new.weights
}

tms.ach.weights = omit.outliers(tms.ach.weights)
eeg.ach.weights = omit.outliers(eeg.ach.weights)

tms.clust.ach = IndependantTest(abs(tms.ach.weights), abs(eeg.ach.weights), testName = "TMS
Cluster Weights Achievers: TMS vs EEG", directionality = "g")
write(capture.output(tms.clust.ach), file = "./Processed Data/EEG Power/CSP/Directional
Tests/Significant Achievers/tmsclust-tTest-sign-Achievers.txt")

plotdat = data.frame(cbind("Weights" = c(abs(tms.ach.weights),

```

```

                                abs(eeg.ach.weights)),
                                "Group" = c(rep("TMS.ach.Weights", length(tms.ach.weights)),
                                rep("EEG.ach.Weights", length(eeg.ach.weights))))))
plotdat$Weights = as.numeric(plotdat$Weights)
gg.plot = ggplot(plotdat, aes(x = Group, y = Weights, fill = Group)) +
  geom_violinhalf(position = position_nudge(x = -.075, y = 0), flip = TRUE) +
  geom_jitter(width = 0.04) +
  geom_boxplot(width = 0.1, position = position_nudge(x = 0.15, y = 0),
    outlier.alpha = 0) +
  theme_light() +
  xlab(label = "TMS and EEG") +
  ylab(label = "CSP Weight") +
  labs(title = "TMS vs EEG Group CSP Weight Comparison, Groups, Achievers") +
  theme(text = element_text(family = "Times")) +
  scale_color_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_fill_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_x_discrete(labels = c("EEG Weights",
                                "TMS Weights")) +
  guides(fill = "none") +
  geom_bracket(inherit.aes = FALSE,
    xmin = "TMS.ach.Weights", xmax = "EEG.ach.Weights",
    y.position = max(plotdat$Weights)+(max(plotdat$Weights)/20), tip.length =
(max(plotdat$Weights)/50),
    label = signif(tms.clust.ach$test$p.value, 2),
    family = "Times")
gg.plot
ggsave(filename = "../Processed Data/EEG Power/CSP/Directional Tests/Significant
Achievers/tmsclust-tTest-sign-Achievers.pdf", width = 6, height = 6)

## TMS Clust NAchievers -----
# creating lists
tms.nach.pids = names(csp.matrices[nach.sign.tms.ind])
tms.nach.weights = NULL
for (pid in tms.nach.pids) {
  new.weights = pid.mean.csp[[pid]][tms.clust.ind]
  start.ind = length(tms.nach.weights) + 1
  end.ind = start.ind + length(new.weights) -1
  tms.nach.weights[start.ind:end.ind] = new.weights
}

# creating lists
eeg.nach.pids = names(csp.matrices[nach.sign.eeg.ind])
eeg.nach.weights = NULL
for (pid in eeg.nach.pids) {
  new.weights = pid.mean.csp[[pid]][tms.clust.ind]
  start.ind = length(eeg.nach.weights) + 1
  end.ind = start.ind + length(new.weights) -1
  eeg.nach.weights[start.ind:end.ind] = new.weights
}

tms.nach.weights = omit.outliers(tms.nach.weights)
eeg.nach.weights = omit.outliers(eeg.nach.weights)

tms.clust.nach = IndependantTest(abs(tms.nach.weights), abs(eeg.nach.weights), testName =
"TMS Cluster Weights Nonachievers: TMS vs EEG", directionality = "t")
write(capture.output(tms.clust.nach), file = "../Processed Data/EEG Power/CSP/Significant
Nonachievers/tmsclust-tTest-sign-nonachievers.txt")

plotdat = data.frame(cbind("Weights" = c(abs(tms.nach.weights),
                                abs(eeg.nach.weights)),
                                "Group" = c(rep("TMS.nach.Weights.Trend",
length(tms.nach.weights)),
                                rep("EEG.nach.Weights.Trend",
length(eeg.nach.weights)))))
plotdat$Weights = as.numeric(plotdat$Weights)
gg.plot = ggplot(plotdat, aes(x = Group, y = Weights, fill = Group)) +
  geom_violinhalf(position = position_nudge(x = -.075, y = 0), flip = TRUE) +

```

```

geom_jitter(width = 0.04) +
geom_boxplot(width = 0.1, position = position_nudge(x = 0.15, y = 0),
             outlier.alpha = 0) +
theme_light() +
xlab(label = "TMS and EEG") +
ylab(label = "CSP Weight") +
labs(title = "TMS vs EEG Group CSP Weight Comparison, Groups, Nonachievers") +
theme(text = element_text(family = "Times")) +
scale_color_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
scale_fill_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
scale_x_discrete(labels = c("EEG Weights",
                           "TMS Weights")) +

guides(fill = "none") +
geom_bracket(inherit.aes = FALSE,
             xmin = "TMS.nach.Weights.Trend", xmax = "EEG.nach.Weights.Trend",
             y.position = max(plotdat$Weights)+(max(plotdat$Weights)/20), tip.length =
(max(plotdat$Weights)/100),
             label = signif(tms.clust.nach$test$p.value, 2),
             family = "Times")
gg.plot
ggsave(filename = "./Processed Data/EEG Power/CSP/Significant Nonachievers/tmsclust-tTest-
sign-nonachievers.pdf", width = 6, height = 6)

## TMS Clust Achievers trend -----
# creating lists
tms.ach.pids = names(csp.matrices[ach.trend.tms.ind])
tms.ach.weights = NULL
for (pid in tms.ach.pids) {
  new.weights = pid.mean.csp[[pid]][tms.clust.ind]
  start.ind = length(tms.ach.weights) + 1
  end.ind = start.ind + length(new.weights) -1
  tms.ach.weights[start.ind:end.ind] = new.weights
}

# creating lists
eeg.ach.pids = names(csp.matrices[ach.trend.eeg.ind])
eeg.ach.weights = NULL
for (pid in eeg.ach.pids) {
  new.weights = pid.mean.csp[[pid]][tms.clust.ind]
  start.ind = length(eeg.ach.weights) + 1
  end.ind = start.ind + length(new.weights) -1
  eeg.ach.weights[start.ind:end.ind] = new.weights
}

tms.ach.weights = omit.outliers(tms.ach.weights)
eeg.ach.weights = omit.outliers(eeg.ach.weights)

tms.clust.ach.trend = IndependantTest(abs(tms.ach.weights), abs(eeg.ach.weights), testName
= "TMS Cluster Weights Achievers Trend: TMS vs EEG", directionality = "t")
write(capture.output(tms.clust.ach.trend), file = "./Processed Data/EEG Power/CSP/Trendlevel
achievers/tmsclust-tTest-Trend-achievers.txt")

plotdat = data.frame(cbind("Weights" = c(abs(tms.ach.weights),
                                         abs(eeg.ach.weights)),
                           "Group" = c(rep("TMS.Ach.Weights.Trend",
length(tms.ach.weights)),
                                         rep("EEG.Ach.Weights.Trend",
length(eeg.ach.weights)))))
plotdat$Weights = as.numeric(plotdat$Weights)
gg.plot = ggplot(plotdat, aes(x = Group, y = Weights, fill = Group)) +
  geom_violinhalf(position = position_nudge(x = -.075, y = 0), flip = TRUE) +
  geom_jitter(width = 0.04) +
  geom_boxplot(width = 0.1, position = position_nudge(x = 0.15, y = 0),
               outlier.alpha = 0) +
  theme_light() +
  xlab(label = "TMS and EEG") +
  ylab(label = "CSP Weight") +

```

```

labs(title = "TMS vs EEG Group CSP Weight Comparison, Groups on Trendlevel, Achievers") +
theme(text = element_text(family = "Times")) +
scale_color_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
scale_fill_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
scale_x_discrete(labels = c("EEG Weights",
                             "TMS Weights")) +

guides(fill = "none") +
geom_bracket(inherit.aes = FALSE,
             xmin = "TMS.Ach.Weights.Trend", xmax = "EEG.Ach.Weights.Trend",
             y.position = max(plotdat$Weights)+(max(plotdat$Weights)/20), tip.length =
(max(plotdat$Weights)/200),
             label = signif(tms.clust.ach.trend$test$p.value, 2),
             family = "Times")

gg.plot
ggsave(filename = "./Processed Data/EEG Power/CSP/Trendlevel Achievers/tmsclust-tTest-Trend-
Achievers.pdf", width = 6, height = 6)

## TMS Clust NAchievers trend -----
# creating lists
tms.nach.pids = names(csp.matrices[nach.trend.tms.ind])
tms.nach.weights = NULL
for (pid in tms.nach.pids) {
  new.weights = pid.mean.csp[[pid]][tms.clust.ind]
  start.ind = length(tms.nach.weights) + 1
  end.ind = start.ind + length(new.weights) -1
  tms.nach.weights[start.ind:end.ind] = new.weights
}

# creating lists
eeg.nach.pids = names(csp.matrices[nach.trend.eeg.ind])
eeg.nach.weights = NULL
for (pid in eeg.nach.pids) {
  new.weights = pid.mean.csp[[pid]][tms.clust.ind]
  start.ind = length(eeg.nach.weights) + 1
  end.ind = start.ind + length(new.weights) -1
  eeg.nach.weights[start.ind:end.ind] = new.weights
}

tms.nach.weights = omit.outliers(tms.nach.weights)
eeg.nach.weights = omit.outliers(eeg.nach.weights)

tms.clust.nach.trend = IndependantTest(abs(tms.nach.weights), abs(eeg.nach.weights),
testName = "TMS Cluster Weights Nonachievers: TMS vs EEG", directionality = "t")
write(capture.output(tms.clust.nach.trend), file = "./Processed Data/EEG
Power/CSP/Trendlevel Nonachievers/tmsclust-tTest-Trend-Nonachievers.txt")

plotdat = data.frame(cbind("Weights" = c(abs(tms.nach.weights),
                                         abs(eeg.nach.weights)),
                           "Group" = c(rep("TMS.Nach.Weights.Trend",
length(tms.nach.weights)),
                                         rep("EEG.Nach.Weights.Trend",
length(eeg.nach.weights)))))
plotdat$Weights = as.numeric(plotdat$Weights)
gg.plot = ggplot(plotdat, aes(x = Group, y = Weights, fill = Group)) +
  geom_violinhalf(position = position_nudge(x = -.075, y = 0), flip = TRUE) +
  geom_jitter(width = 0.04) +
  geom_boxplot(width = 0.1, position = position_nudge(x = 0.15, y = 0),
              outlier.alpha = 0) +
  theme_light() +
  xlab(label = "TMS and EEG") +
  ylab(label = "CSP Weight") +
  labs(title = "TMS vs EEG Group CSP Weight Comparison, Groups on Trendlevel,
Nonachievers") +
  theme(text = element_text(family = "Times")) +
  scale_color_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_fill_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +

```

```

scale_x_discrete(labels = c("EEG Weights",
                             "TMS Weights")) +

guides(fill = "none") +
geom_bracket(inherit.aes = FALSE,
             xmin = "TMS.Nach.Weights.Trend", xmax = "EEG.Nach.Weights.Trend",
             y.position = max(plotdat$Weights)+(max(plotdat$Weights)/20), tip.length =
(max(plotdat$Weights)/100),
             label = signif(tms.clust.nach.trend$test$p.value, 2),
             family = "Times")

gg.plot
ggsave(filename = "../Processed Data/EEG Power/CSP/Trendlevel Nonachievers/tmsclust-tTest-
Trend-Nonachievers.pdf", width = 6, height = 6)

## TMS Clust Achievers -----
# creating lists
ach.pids = names(csp.matrices[ach.sign.ind])
ach.weights = NULL
for (pid in ach.pids) {
  new.weights = pid.mean.csp[[pid]][tms.clust.ind]
  start.ind = length(ach.weights) + 1
  end.ind = start.ind + length(new.weights) -1
  ach.weights[start.ind:end.ind] = new.weights
}

# creating lists
nach.pids = names(csp.matrices[nach.sign.ind])
nach.weights = NULL
for (pid in nach.pids) {
  new.weights = pid.mean.csp[[pid]][tms.clust.ind]
  start.ind = length(nach.weights) + 1
  end.ind = start.ind + length(new.weights) -1
  nach.weights[start.ind:end.ind] = new.weights
}

ach.weights = omit.outliers(ach.weights)
nach.weights = omit.outliers(nach.weights)

tms.clust.ach = IndependentTest(abs(ach.weights), abs(nach.weights), testName = "TMS
Cluster Weights Achievers vs Nonachievers", directionality = "t")
write(capture.output(tms.clust.ach), file = "../Processed Data/EEG
Power/CSP/Achievers/tmsclust-tTest-Achievers.txt")

plotdat = data.frame(cbind("Weights" = c(abs(ach.weights),
                                         abs(nach.weights)),
                           "Control" = c(rep("Ach.Weights.Trend", length(ach.weights)),
                                         rep("Nach.Weights.Trend", length(nach.weights)))))
plotdat$Weights = as.numeric(plotdat$Weights)
gg.plot = ggplot(plotdat, aes(x = Control, y = Weights, fill = Control)) +
  geom_violinhalf(position = position_nudge(x = -.075, y = 0), flip = TRUE) +
  geom_jitter(width = 0.04) +
  geom_boxplot(width = 0.1, position = position_nudge(x = 0.15, y = 0),
              outlier.alpha = 0) +
  theme_light() +
  # xlab(label = "TMS and EEG") +
  ylab(label = "CSP Weight") +
  labs(title = "CSP Weight Comparison Achievers vs Nonachievers") +
  theme(text = element_text(family = "Times")) +
  scale_color_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_fill_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_x_discrete(labels = c("Achiever Weights",
                              "Nonachiever Weights")) +

guides(fill = "none") +
geom_bracket(inherit.aes = FALSE,
             xmin = "Ach.Weights.Trend", xmax = "Nach.Weights.Trend",
             y.position = max(plotdat$Weights)+(max(plotdat$Weights)/20), tip.length =
(max(plotdat$Weights)/200),
             label = signif(tms.clust.ach$test$p.value, 2),

```

```

        family = "Times")
gg.plot
ggsave(filename = "../Processed Data/EEG Power/CSP/Achievers/tmsclust-tTest-Achievers.pdf",
width = 6, height = 6)

# EEG Clust Achievers -----
# creating lists
tms.ach.pids = names(csp.matrices[ach.sign.tms.ind])
tms.ach.weights = NULL
for (pid in tms.ach.pids) {
  new.weights = pid.mean.csp[[pid]][eeg.clust.ind]
  start.ind = length(tms.ach.weights) + 1
  end.ind = start.ind + length(new.weights) -1
  tms.ach.weights[start.ind:end.ind] = new.weights
}

# creating lists
eeg.ach.pids = names(csp.matrices[ach.sign.eeg.ind])
eeg.ach.weights = NULL
for (pid in eeg.ach.pids) {
  new.weights = pid.mean.csp[[pid]][eeg.clust.ind]
  start.ind = length(eeg.ach.weights) + 1
  end.ind = start.ind + length(new.weights) -1
  eeg.ach.weights[start.ind:end.ind] = new.weights
}

tms.ach.weights = omit.outliers(tms.ach.weights)
eeg.ach.weights = omit.outliers(eeg.ach.weights)

eeg.clust.ach = IndependantTest(abs(tms.ach.weights), abs(eeg.ach.weights), testName = "EEG
Cluster Weights Achievers: TMS vs EEG", directionality = "t")
write(capture.output(eeg.clust.ach), file = "../Processed Data/EEG Power/CSP/Significant
Achievers/eegclust-tTest-sign-Achievers.txt")

plotdat = data.frame(cbind("Weights" = c(abs(tms.ach.weights),
                                         abs(eeg.ach.weights)),
                           "Group" = c(rep("TMS.ach.Weights", length(tms.ach.weights)),
                                         rep("EEG.ach.Weights", length(eeg.ach.weights)))))
plotdat$Weights = as.numeric(plotdat$Weights)
gg.plot = ggplot(plotdat, aes(x = Group, y = Weights, fill = Group)) +
  geom_violinhalf(position = position_nudge(x = -.075, y = 0), flip = TRUE) +
  geom_jitter(width = 0.04) +
  geom_boxplot(width = 0.1, position = position_nudge(x = 0.15, y = 0),
              outlier.alpha = 0) +
  theme_light() +
  xlab(label = "TMS and EEG") +
  ylab(label = "CSP Weight") +
  labs(title = "TMS vs EEG Group CSP Weight Comparison, Groups, Achievers") +
  theme(text = element_text(family = "Times")) +
  scale_color_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_fill_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_x_discrete(labels = c("EEG Weights",
                              "TMS Weights")) +
  guides(fill = "none") +
  geom_bracket(inherit.aes = FALSE,
              xmin = "TMS.ach.Weights", xmax = "EEG.ach.Weights",
              y.position = max(plotdat$Weights)+(max(plotdat$Weights)/20), tip.length =
(max(plotdat$Weights)/100),
              label = signif(eeg.clust.ach$test$p.value, 2),
              family = "Times")
gg.plot
ggsave(filename = "../Processed Data/EEG Power/CSP/Significant Achievers/eegclust-tTest-sign-
Achievers.pdf", width = 6, height = 6)

## EEG Clust NAchievers -----
# creating lists

```

```

tms.nach.pids = names(csp.matrices[nach.sign.tms.ind])
tms.nach.weights = NULL
for (pid in tms.nach.pids) {
  new.weights = pid.mean.csp[[pid]][eeg.clust.ind]
  start.ind = length(tms.nach.weights) + 1
  end.ind = start.ind + length(new.weights) -1
  tms.nach.weights[start.ind:end.ind] = new.weights
}

# creating lists
eeg.nach.pids = names(csp.matrices[nach.sign.eeg.ind])
eeg.nach.weights = NULL
for (pid in eeg.nach.pids) {
  new.weights = pid.mean.csp[[pid]][eeg.clust.ind]
  start.ind = length(eeg.nach.weights) + 1
  end.ind = start.ind + length(new.weights) -1
  eeg.nach.weights[start.ind:end.ind] = new.weights
}

tms.nach.weights = omit.outliers(tms.nach.weights)
eeg.nach.weights = omit.outliers(eeg.nach.weights)

eeg.clust.nach = IndependantTest(abs(tms.nach.weights), abs(eeg.nach.weights), testName =
"EEG Cluster Weights Nonachievers: TMS vs EEG", directionality = "t")
write(capture.output(eeg.clust.nach), file = "./Processed Data/EEG Power/CSP/Significant
Nonachievers/eegclust-tTest-sign-nonachievers.txt")

plotdat = data.frame(cbind("Weights" = c(abs(tms.nach.weights),
                                         abs(eeg.nach.weights)),
                           "Group" = c(rep("TMS.nach.Weights.Trend",
length(tms.nach.weights)),
                                         rep("EEG.nach.Weights.Trend",
length(eeg.nach.weights)))))
plotdat$Weights = as.numeric(plotdat$Weights)
gg.plot = ggplot(plotdat, aes(x = Group, y = Weights, fill = Group)) +
  geom_violinhalf(position = position_nudge(x = -.075, y = 0), flip = TRUE) +
  geom_jitter(width = 0.04) +
  geom_boxplot(width = 0.1, position = position_nudge(x = 0.15, y = 0),
               outlier.alpha = 0) +
  theme_light() +
  xlab(label = "TMS and EEG") +
  ylab(label = "CSP Weight") +
  labs(title = "TMS vs EEG Group CSP Weight Comparison, Groups, Nonachievers") +
  theme(text = element_text(family = "Times")) +
  scale_color_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_fill_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_x_discrete(labels = c("EEG Weights",
                              "TMS Weights")) +
  guides(fill = "none") +
  geom_bracket(inherit.aes = FALSE,
               xmin = "TMS.nach.Weights.Trend", xmax = "EEG.nach.Weights.Trend",
               y.position = max(plotdat$Weights)+(max(plotdat$Weights)/20), tip.length =
(max(plotdat$Weights)/100),
               label = signif(eeg.clust.nach$test$p.value, 2),
               family = "Times")
gg.plot
ggsave(filename = "./Processed Data/EEG Power/CSP/Significant Nonachievers/eegclust-tTest-
sign-nonachievers.pdf", width = 6, height = 6)

## EEG Clust Achievers trend -----
# creating lists
tms.ach.pids = names(csp.matrices[ach.trend.tms.ind])
tms.ach.weights = NULL
for (pid in tms.ach.pids) {
  new.weights = pid.mean.csp[[pid]][eeg.clust.ind]
  start.ind = length(tms.ach.weights) + 1
  end.ind = start.ind + length(new.weights) -1

```



```

tms.ach.weights[start.ind:end.ind] = new.weights
}

# creating lists
eeg.ach.pids = names(csp.matrices[ach.trend.eeg.ind])
eeg.ach.weights = NULL
for (pid in eeg.ach.pids) {
  new.weights = pid.mean.csp[[pid]][eeg.clust.ind]
  start.ind = length(eeg.ach.weights) + 1
  end.ind = start.ind + length(new.weights) - 1
  eeg.ach.weights[start.ind:end.ind] = new.weights
}

tms.ach.weights = omit.outliers(tms.ach.weights)
eeg.ach.weights = omit.outliers(eeg.ach.weights)

eeg.clust.ach.trend = IndependantTest(abs(tms.ach.weights), abs(eeg.ach.weights), testName
= "EEG Cluster Weights Achievers Trend: TMS vs EEG", directionality = "t")
write(capture.output(eeg.clust.ach.trend), file = "./Processed Data/EEG Power/CSP/Trendlevel
achievers/eegclust-tTest-Trend-achievers.txt")

plotdat = data.frame(cbind("Weights" = c(abs(tms.ach.weights),
                                     abs(eeg.ach.weights)),
                           "Group" = c(rep("TMS.Ach.Weights.Trend",
length(tms.ach.weights)),
                                     rep("EEG.Ach.Weights.Trend",
length(eeg.ach.weights)))))
plotdat$Weights = as.numeric(plotdat$Weights)
gg.plot = ggplot(plotdat, aes(x = Group, y = Weights, fill = Group)) +
  geom_violinhalf(position = position_nudge(x = -.075, y = 0), flip = TRUE) +
  geom_jitter(width = 0.04) +
  geom_boxplot(width = 0.1, position = position_nudge(x = 0.15, y = 0),
               outlier.alpha = 0) +
  theme_light() +
  xlab(label = "TMS and EEG") +
  ylab(label = "CSP Weight") +
  labs(title = "TMS vs EEG Group CSP Weight Comparison, Groups on Trendlevel, Achievers") +
  theme(text = element_text(family = "Times")) +
  scale_color_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_fill_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_x_discrete(labels = c("EEG Weights",
                              "TMS Weights")) +
  guides(fill = "none") +
  geom_bracket(inherit.aes = FALSE,
               xmin = "TMS.Ach.Weights.Trend", xmax = "EEG.Ach.Weights.Trend",
               y.position = max(plotdat$Weights)+(max(plotdat$Weights)/20), tip.length =
(max(plotdat$Weights)/100),
               label = signif(eeg.clust.ach.trend$test$p.value, 2),
               family = "Times")
gg.plot
ggsave(filename = "./Processed Data/EEG Power/CSP/Trendlevel Achievers/eegclust-tTest-Trend-
Achievers.pdf", width = 6, height = 6)

## EEG Clust NAchievers trend -----
# creating lists
tms.nach.pids = names(csp.matrices[nach.trend.tms.ind])
tms.nach.weights = NULL
for (pid in tms.nach.pids) {
  new.weights = pid.mean.csp[[pid]][eeg.clust.ind]
  start.ind = length(tms.nach.weights) + 1
  end.ind = start.ind + length(new.weights) - 1
  tms.nach.weights[start.ind:end.ind] = new.weights
}

# creating lists
eeg.nach.pids = names(csp.matrices[nach.trend.eeg.ind])

```



```

eeg.nach.weights = NULL
for (pid in eeg.nach.pids) {
  new.weights = pid.mean.csp[[pid]][eeg.clust.ind]
  start.ind = length(eeg.nach.weights) + 1
  end.ind = start.ind + length(new.weights) - 1
  eeg.nach.weights[start.ind:end.ind] = new.weights
}

tms.nach.weights = omit.outliers(tms.nach.weights)
eeg.nach.weights = omit.outliers(eeg.nach.weights)

eeg.clust.nach.trend = IndependentTest(abs(tms.nach.weights), abs(eeg.nach.weights),
testName = "EEG Cluster Weights Nonachievers: TMS vs EEG", directionality = "t")
write(capture.output(eeg.clust.nach.trend), file = "./Processed Data/EEG
Power/CSP/Trendlevel Nonachievers/eegclust-tTest-Trend-Nonachievers.txt")

plotdat = data.frame(cbind("Weights" = c(abs(tms.nach.weights),
                                     abs(eeg.nach.weights)),
                           "Group" = c(rep("TMS.Nach.Weights.Trend",
length(tms.nach.weights)),
                                     rep("EEG.Nach.Weights.Trend",
length(eeg.nach.weights)))))
plotdat$Weights = as.numeric(plotdat$Weights)
gg.plot = ggplot(plotdat, aes(x = Group, y = Weights, fill = Group)) +
  geom_violinhalf(position = position_nudge(x = -.075, y = 0), flip = TRUE) +
  geom_jitter(width = 0.04) +
  geom_boxplot(width = 0.1, position = position_nudge(x = 0.15, y = 0),
               outlier.alpha = 0) +
  theme_light() +
  xlab(label = "TMS and EEG") +
  ylab(label = "CSP Weight") +
  labs(title = "TMS vs EEG Group CSP Weight Comparison, Groups on Trendlevel,
Nonachievers") +
  theme(text = element_text(family = "Times")) +
  scale_color_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_fill_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_x_discrete(labels = c("EEG Weights",
                              "TMS Weights")) +
  guides(fill = "none") +
  geom_bracket(inherit.aes = FALSE,
               xmin = "TMS.Nach.Weights.Trend", xmax = "EEG.Nach.Weights.Trend",
               y.position = max(plotdat$Weights)+(max(plotdat$Weights)/20), tip.length =
(max(plotdat$Weights)/100),
               label = signif(eeg.clust.nach.trend$test$p.value, 2),
               family = "Times")
gg.plot
ggsave(filename = "./Processed Data/EEG Power/CSP/Trendlevel Nonachievers/eegclust-tTest-
Trend-Nonachievers.pdf", width = 6, height = 6)

## EEG Cluster Achievers -----
# creating lists
ach.pids = names(csp.matrices[ach.sign.ind])
ach.weights = NULL
for (pid in ach.pids) {
  new.weights = pid.mean.csp[[pid]][eeg.clust.ind]
  start.ind = length(ach.weights) + 1
  end.ind = start.ind + length(new.weights) - 1
  ach.weights[start.ind:end.ind] = new.weights
}
#ach.weights = data.frame("Weights" = ach.weights)

# creating lists
nach.pids = names(csp.matrices[nach.sign.ind])
nach.weights = NULL
for (pid in nach.pids) {
  new.weights = pid.mean.csp[[pid]][eeg.clust.ind]
  start.ind = length(nach.weights) + 1

```

```

    end.ind = start.ind + length(new.weights) -1
    nach.weights[start.ind:end.ind] = new.weights
  }
#nach.weights = data.frame("Weights" = nach.weights)

# nach.weights$outliers = Check_For_Outliers(chunk.data.y = nach.weights$Weights,
pdf.path.name = "./Processed Data/EEG Power/CSP/Achievers/eegclust-nach-
outliers.pdf")$outliers
# ach.weights$outliers = Check_For_Outliers(chunk.data.y = ach.weights$Weights,
pdf.path.name = "./Processed Data/EEG Power/CSP/Achievers/eegclust-ach-
outliers.pdf")$outliers

# omit outliers?
nach.weights = omit.outliers(nach.weights)
ach.weights = omit.outliers(ach.weights)

eeg.clust.ach = IndependantTest(abs(ach.weights),
                                abs(nach.weights), testName = "EEG Cluster Weights
Achievers vs Nonachievers", directionality = "t")
write(capture.output(eeg.clust.ach), file = "./Processed Data/EEG
Power/CSP/Achievers/eegclust-tTest-Achievers.txt")

plotdat = data.frame(cbind("Weights" = c(abs(ach.weights),
                                abs(nach.weights)),
                            "Control" = c(rep("Ach.Weights.Trend", length(ach.weights)),
                                rep("Nach.Weights.Trend", length(nach.weights)))))
plotdat$Weights = as.numeric(plotdat$Weights)
gg.plot = ggplot(plotdat, aes(x = Control, y = Weights, fill = Control)) +
  geom_violinhalf(position = position_nudge(x = -.075, y = 0), flip = TRUE) +
  geom_jitter(width = 0.04) +
  geom_boxplot(width = 0.1, position = position_nudge(x = 0.15, y = 0),
    outlier.alpha = 0) +
  theme_light() +
  # xlab(label = "TMS and EEG") +
  ylab(label = "CSP Weight") +
  labs(title = "CSP Weight Comparison Achievers vs Nonachievers") +
  theme(text = element_text(family = "Times")) +
  scale_color_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_fill_viridis_d(alpha = 0.8, begin = 0.2, end = 0.8) +
  scale_x_discrete(labels = c("Achiever Weights",
                              "Nonachiever Weights")) +
  guides(fill = "none") +
  geom_bracket(inherit.aes = FALSE,
    xmin = "Ach.Weights.Trend", xmax = "Nach.Weights.Trend",
    y.position = max(plotdat$Weights)+(max(plotdat$Weights)/20), tip.length =
(max(plotdat$Weights)/100),
    label = signif(eeg.clust.ach$test$p.value, 2),
    family = "Times")
gg.plot
ggsave(filename = "./Processed Data/EEG Power/CSP/Achievers/eegclust-tTest-Achievers.pdf",
width = 6, height = 6)

```

10.4.13 MEP Hypothesis Testing (R)

This script brings together all the MEP data and tests the preregistered hypotheses.

```
# This script we will Bring together the files

# Init -----
# clear all
rm(list = ls())

# Libraries
library(lmerTest)
library(lme4)
library(doParallel)
library(ggplot2)
library(extrafont)
library(ggpubr)
library(see)

# set wd
work_dir = ""
setwd(work_dir)

# get useful functions
source(file = "../../../Code/R Code/R_useful_funcs_share.R")

# Start Parallel core -----

# Load Data -----

# Load MEP Data
load(file = "")

# Load QS Data
load(file = "")

# dataframe with just up and down
df.updown = results[results$conditions %in% c("up", "down"),]

# exclude background rMS higher than 0.01
df.updown = df.updown[df.updown$actual_rMS_right < 0.01, ]

# ztransform rms, because it's small scale is causing issues
df.updown$ztrans.rms = (df.updown$actual_rMS_right -
mean(df.updown$actual_rMS_right))/sd(df.updown$actual_rMS_right)

# exclude outliers in rms and mep
df.updown = df.updown[!ID.outliers(df.updown$actual_rMS_right),]
df.updown = df.updown[!ID.outliers(df.updown$actual_MEP),]

# Remove Sess 1 and 2 from the EEG dataset
# Create group variable for CA scores:
TMS.PIDs = c('')
df.merged$Group = "EEG"
df.merged$Group[df.merged$PID %in% TMS.PIDs] = "TMS"
df.merged$Continuous.Block = (df.merged$Sess - 1) * 4 + df.merged$Block

df.merged$CA = as.numeric(df.merged$CA)
df.merged$`KVIQ Total` = as.numeric(df.merged$`KVIQ Total`)
df.merged$`KVIQ Imagery` = as.numeric(df.merged$`KVIQ Imagery`)
df.merged$`KVIQ Intensity` = as.numeric(df.merged$`KVIQ Intensity`)
df.merged$KVIQ.Intensity = df.merged$`KVIQ Intensity`

df.eegbci = df.merged
```

```

df.eegbci = df.eegbci[df.eegbci$Sess %in% c(3, 4, 5), ]
# remove block 0 (cal block)
df.eegbci = df.eegbci[!df.eegbci$Block == 0, ]

# # Hypothesis 1 -----
## Looking at mixed effects in the null model -----

n0.m1 = nullmodel_regression_function(df.updown, actual_MEP ~ 1, 'pid')
n0.m2 = nullmodel_regression_function(df.updown, Dynamic.Standardised.MEP ~ 1, 'pid')
# turns out the best is probably the (1|pid)

# save outputs:
save(n0.m1, file = paste0(work_dir, './Processed Data/MEP Data/Model Data/M0/m0-actMEP.rda'))
write(capture.output(summary(n0.m1)), file = paste0(work_dir, './Processed Data/MEP Data/Model Data/M0/m0-actMEP-outp.txt'))
save(n0.m2, file = paste0(work_dir, './Processed Data/MEP Data/Model Data/M0/m0-dynMEP.rda'))
write(capture.output(summary(n0.m2)), file = paste0(work_dir, './Processed Data/MEP Data/Model Data/M0/m0-dynMEP-outp.txt'))
## Looking at main effects in Model -----

m1.1 = lmer(Dynamic.Standardised.MEP ~ conditions + (1|pid), data = df.updown)
m1.2 = lmer(Dynamic.Standardised.MEP ~ mep.nr.train + (1|pid), data = df.updown)
summary(m1.1)
summary(m1.2)
anova(m1.1, m1.2)

# full model
m2.1 = lmer(Dynamic.Standardised.MEP ~ conditions + mep.nr.train + (1|pid), data = df.updown)
m2.2.act = fourcore_lmer(actual_MEP ~ conditions * mep.nr.train + (1|pid), data = df.updown)
m2.2.dyn = fourcore_lmer(Dynamic.Standardised.MEP ~ conditions * mep.nr.train + (1|pid), data = df.updown)
save(m2.2.act, file = paste0(work_dir, './Processed Data/MEP Data/Model Data/Mnorslo/Mnorslo-actMEP.rda'))
write(capture.output(summary(m2.2.act)), file = paste0(work_dir, './Processed Data/MEP Data/Model Data/Mnorslo/Mnorslo-actMEP-outp.txt'))
save(m2.2.dyn, file = paste0(work_dir, './Processed Data/MEP Data/Model Data/Mnorslo/Mnorslo-dynMEP.rda'))
write(capture.output(summary(m2.2.dyn)), file = paste0(work_dir, './Processed Data/MEP Data/Model Data/Mnorslo/Mnorslo-dynMEP-outp.txt'))
comp.Mnorslo = anova(m2.2.act, m2.2.dyn)
write(capture.output(comp.Mnorslo), file = paste0(work_dir, './Processed Data/MEP Data/Model Data/Mnorslo/Mnorslo-comp.txt'))

# Model including Session
m2.3.act = fourcore_lmer(actual_MEP ~ conditions * mep.nr.train + sess + (1|pid), data = df.updown)
m2.4.dyn = fourcore_lmer(Dynamic.Standardised.MEP ~ conditions * mep.nr.train + sess + (1|pid), data = df.updown)
save(m2.3.act, file = paste0(work_dir, './Processed Data/MEP Data/Model Data/MSess/MSess-actMEP.rda'))
write(capture.output(summary(m2.3.act)), file = paste0(work_dir, './Processed Data/MEP Data/Model Data/MSess/MSess-actMEP-outp.txt'))
save(m2.4.dyn, file = paste0(work_dir, './Processed Data/MEP Data/Model Data/MSess/MSess-dynMEP.rda'))
write(capture.output(summary(m2.4.dyn)), file = paste0(work_dir, './Processed Data/MEP Data/Model Data/MSess/MSess-dynMEP-outp.txt'))

# Model including eeg & Sess
m5.3.act = fourcore_lmer(actual_MEP ~ conditions * mep.nr.train + sess + eeg.condition + (1|pid), data = df.updown)
m5.4.dyn = fourcore_lmer(Dynamic.Standardised.MEP ~ conditions * mep.nr.train + sess + eeg.condition + (1|pid), data = df.updown)
summary(m5.3.act)

```

```

summary(m5.4.dyn)

# save this model
save(m5.3.act, file = paste0(work_dir, './Processed Data/MEP Data/Model
Data/MSessInt/MSessInt-actMEP.rda'))
write(capture.output(summary(m5.3.act)), file = paste0(work_dir, './Processed Data/MEP
Data/Model Data/MSessInt/MSessInt-actMEP-outp.txt'))
save(m5.4.dyn, file = paste0(work_dir, './Processed Data/MEP Data/Model
Data/MSessInt/MSessInt-dynMEP.rda'))
write(capture.output(summary(m5.4.dyn)), file = paste0(work_dir, './Processed Data/MEP
Data/Model Data/MSessInt/MSessInt-dynMEP-outp.txt'))
write(capture.output(anova(m5.3.act, m2.3.act, m2.2.act)), file = paste0(work_dir,
'./Processed Data/MEP Data/Model Data/MSessInt/MSessInt-actMEP-comp-outp.txt'))
write(capture.output(anova(m5.4.dyn, m2.4.dyn, m2.2.dyn)), file = paste0(work_dir,
'./Processed Data/MEP Data/Model Data/MSessInt/MSessInt-dynMEP-comp-outp.txt'))

summary(m2.1)
summary(m2.2)
anova(m2.1, m2.2, m1.1, m1.2)

h1.m1.dyn = add2model_regression_function(df.updown,
                                         Dynamic.Standardised.MEP ~ 1 + (1|pid),
                                         "conditions",
                                         "mep.nr.train")

h1.m1.raw = add2model_regression_function(df.updown,
                                         actual_MEP ~ 1 + (1|pid),
                                         "conditions",
                                         "mep.nr.train")

m3.1.dyn = fourcore_lmer(Dynamic.Standardised.MEP ~ conditions * mep.nr.train + (pid|pid),
data = df.updown)
m3.1.act = fourcore_lmer(actual_MEP ~ conditions * mep.nr.train + (pid|pid), data =
df.updown)
save(m3.1.act, file = paste0(work_dir, './Processed Data/MEP Data/Model Data/MFull/MFull-
actMEP.rda'))
write(capture.output(summary(m3.1.act)), file = paste0(work_dir, './Processed Data/MEP
Data/Model Data/MFull/MFull-actMEP-outp.txt'))
save(m3.1.dyn, file = paste0(work_dir, './Processed Data/MEP Data/Model Data/MFull/MFull-
dynMEP.rda'))
write(capture.output(summary(m3.1.dyn)), file = paste0(work_dir, './Processed Data/MEP
Data/Model Data/MFull/MFull-dynMEP-outp.txt'))
comp.Mfull = anova(m3.1.act, m3.1.dyn, m5.3.act, )
write(capture.output(comp.Mfull), file = paste0(work_dir, './Processed Data/MEP Data/Model
Data/MFull/MFull-comp.txt'))

m3.2 = fourcore_lmer(Dynamic.Standardised.MEP ~ conditions * mep.nr.train + ztrans.rms +
(1|pid), data = df.updown)
summary(m3.2)

# Check about background rMS
m.rMS.act = fourcore_lmer(actual_MEP ~ conditions * mep.nr.train + ztrans.rms + (1|pid),
data = df.updown)
Model_me_this(model.data = df.updown,
               model.formula = "actual_MEP ~ conditions * mep.nr.train * ztrans.rms +
(1|pid)",
               save_path = "./Processed Data/MEP Data/rMS/rMS-int",
               plot.model = FALSE)

m.rMS.dyn = fourcore_lmer(Dynamic.Standardised.MEP ~ conditions * mep.nr.train + ztrans.rms
+ (1|pid), data = df.updown)
Model_me_this(model.data = df.updown,
               model.formula = "Dynamic.Standardised.MEP ~ conditions * mep.nr.train *
ztrans.rms + (1|pid)",
               save_path = "./Processed Data/MEP Data/rMS-dynmep/rMS-int-dynmep",
               plot.model = FALSE)

m.rMS.int.act = fourcore_lmer(actual_MEP ~ conditions * mep.nr.train + ztrans.rms +

```

```

                                conditions:ztrans.rms + (1|pid), data = df.updown)
m.rMS.int.dyn = fourcore_lmer(Dynamic.Standardised.MEP ~ conditions * mep.nr.train *
ztrans.rms + (1|pid), data = df.updown)
summary(m.rMS.int.act)
summary(m.rMS.int.dyn)
# save rMS results
save(m.rMS.act, file = paste0(work_dir, './Processed Data/MEP Data/Model Data/MrMS/MrMS-
actMEP.rda'))
write(capture.output(summary(m.rMS.act)), file = paste0(work_dir, './Processed Data/MEP
Data/Model Data/MrMS/MrMS-actMEP-outp.txt'))
save(m.rMS.int.act, file = paste0(work_dir, './Processed Data/MEP Data/Model
Data/MrMS/MrMSInt-actMEP.rda'))
write(capture.output(summary(m.rMS.int.act)), file = paste0(work_dir, './Processed Data/MEP
Data/Model Data/MrMS/MrMSInt-actMEP-outp.txt'))
comp.MrMS = anova(m.rMS.act, m.rMS.int.act, m2.2.act, m5.3.act)
write(capture.output(comp.MrMS), file = paste0(work_dir, './Processed Data/MEP Data/Model
Data/MrMS/MrMS-act-comp.txt'))
save(m.rMS.dyn, file = paste0(work_dir, './Processed Data/MEP Data/Model Data/MrMS/MrMS-
dynMEP.rda'))
write(capture.output(summary(m.rMS.dyn)), file = paste0(work_dir, './Processed Data/MEP
Data/Model Data/MrMS/MrMS-dynMEP-outp.txt'))
save(m.rMS.int.dyn, file = paste0(work_dir, './Processed Data/MEP Data/Model
Data/MrMS/MrMSInt-dynMEP.rda'))
write(capture.output(summary(m.rMS.int.dyn)), file = paste0(work_dir, './Processed Data/MEP
Data/Model Data/MrMS/MrMSInt-dynMEP-outp.txt'))
comp.MrMS = anova(m.rMS.dyn, m.rMS.int.dyn, m5.4.dyn, m2.2.dyn)
write(capture.output(comp.MrMS), file = paste0(work_dir, './Processed Data/MEP Data/Model
Data/MrMS/MrMS-dyn-comp.txt'))

# Most complete model using all conditions and rms
m.comp.act = fourcore_lmer(actual_MEP ~ conditions * mep.nr.train + ztrans.rms +
                                conditions:ztrans.rms + sess + eeg.condition + (1|pid), data =
df.updown)
m.comp.dyn = fourcore_lmer(Dynamic.Standardised.MEP ~ conditions * mep.nr.train +
ztrans.rms +
                                conditions:ztrans.rms + sess + eeg.condition + (1|pid), data
= df.updown)
save(m.comp.dyn, file = paste0(work_dir, './Processed Data/MEP Data/Model
Data/MComp/MCompInt-dynMEP.rda'))
write(capture.output(summary(m.comp.dyn)), file = paste0(work_dir, './Processed Data/MEP
Data/Model Data/MComp/MCompInt-dynMEP-outp.txt'))
comp.MComp = anova(m.comp.dyn, m.rMS.dyn, m.rMS.int.dyn, m5.4.dyn, m2.2.dyn)
write(capture.output(comp.MComp), file = paste0(work_dir, './Processed Data/MEP Data/Model
Data/MComp/MComp-dyn-comp.txt'))
save(m.comp.act, file = paste0(work_dir, './Processed Data/MEP Data/Model
Data/MComp/MCompInt-actMEP.rda'))
write(capture.output(summary(m.comp.act)), file = paste0(work_dir, './Processed Data/MEP
Data/Model Data/MComp/MCompInt-actMEP-outp.txt'))
comp.MComp = anova(m.comp.act, m.rMS.act, m.rMS.int.act, m2.2.act, m5.3.act)
write(capture.output(comp.MComp), file = paste0(work_dir, './Processed Data/MEP Data/Model
Data/MComp/MComp-act-comp.txt'))

# Other rMS Measures -----

# reverse Model
rms.simple = fourcore_lmer(actual_rMS_right ~ conditions * mep.nr.train +
Dynamic.Standardised.MEP + (1|pid), data = df.updown)
summary(rms.simple)
rms.simple = fourcore_lmer(ztrans.rms ~ conditions * mep.nr.train +
Dynamic.Standardised.MEP + (1|pid), data = df.updown)
summary(rms.simple)
rms.full = fourcore_lmer(ztrans.rms ~ conditions * mep.nr.train * Dynamic.Standardised.MEP
+
                                sess + eeg.condition + (1|pid), data = df.updown)
summary(rms.full)

# reverse Model with log?

```

```

df.updown$log.rms = log10(df.updown$actual_rMS_right)
rms.simple = fourcore_lmer(log.rms ~ conditions * mep.nr.train + actual_MEP + (1|pid), data = df.updown)
summary(rms.simple)
rms.full = fourcore_lmer(log.rms ~ conditions * mep.nr.train * actual_MEP + sess + eeg.condition + (1|pid), data = df.updown)
summary(rms.full)

test.model = lmer(actual_MEP ~ conditions * mep.nr.train + ztrans.rms + (1|pid), data = df.updown)
summary(test.model)

# H2 -----

# PID 51 seems to be missing, so let's exclude NA them for the moment
missing = df.eegbci[is.na(df.eegbci$CA),]
df.eegbci = df.eegbci[!is.na(df.eegbci$CA),]

H2.n0 = nullmodel_regression_function(df.eegbci, CA ~ 1, 'PID')
# so we should be using (1|PID)

H2.m1 = add2model_regression_function(df.eegbci, CA ~ 1 + (1|PID), "Group", "Continuous.Block")
H2.m2 = add2model_regression_function(df.eegbci, CA ~ 1 + (Group|PID), "Group", "Continuous.Block")

# compare two models
h2.m1.1 = lmer(CA ~ Group * Continuous.Block + (1|PID), data = df.eegbci)
h2.m2.1 = lmer(CA ~ Group * Continuous.Block + (Group|PID), data = df.eegbci)
summary(h2.m1.1)
summary(h2.m2.1)
anova(h2.m1.1, h2.m2.1)
# Group|PID not much better

# how about adding Mental imagery scores and kss?
# checking out kss
H2.m1.kviql.tot = lmer(CA ~ Group * Continuous.Block * `KVIQ Total` + (1|PID), data = df.eegbci)
H2.m1.kviql.vis = lmer(CA ~ Group * Continuous.Block * `KVIQ Imagery` + (1|PID), data = df.eegbci)
H2.m1.kviql.int = lmer(CA ~ Group * Continuous.Block * `KVIQ Intensity` + (1|PID), data = df.eegbci)
summary(H2.m1.kviql.tot)
summary(H2.m1.kviql.vis)
summary(H2.m1.kviql.int)
anova(H2.m1.kviql.tot, H2.m1.kviql.vis, H2.m1.kviql.int)

H2.m1 = add2model_regression_function(df.eegbci, CA ~ 1 + Group * Continuous.Block + (1|PID), "KSS", "KVIQ.Intensity")
# very interesting KSS makes model better! KVIQ not so much!
# compare both models, although I have to exclude KSS!
df.kss = df.eegbci[!is.na(df.eegbci$KSS),]
H2.m1.KSS = lmer(CA ~ Group * Continuous.Block + KSS + (1|PID), data = df.kss)
H2.m2.KSS = lmer(CA ~ Group * Continuous.Block + KSS + KSS:Group + (1|PID), data = df.kss)
H2.m3.KSS = lmer(CA ~ Group * Continuous.Block * KSS + (1|PID), data = df.kss)
summary(H2.m2.KSS)
summary(H2.m3.KSS)
anova(H2.m1.KSS, H2.m2.KSS, H2.m3.KSS)
H2.m1.noKSS = lmer(CA ~ Group * Continuous.Block + (1|PID), data = df.kss)
H2.m1.Sess = lmer(CA ~ Group * Continuous.Block + Sess + (1|PID), data = df.kss)
summary(H2.m1.KSS)
summary(H2.m1.noKSS)
summary(H2.m1.Sess)
anova(H2.m1.KSS, H2.m1.noKSS, H2.m1.Sess)

# just looking at failure bias for fun
FB.m1 = lmer(`Failure Bias` ~ Group * Continuous.Block + Sess + (1|PID), data = df.kss)
summary(FB.m1)

```



```

# MEP & KVIQ? -----
df.mepkviq = df.updown
df.mepkviq$kviq.total = 0
df.mepkviq$kviq.imagery = 0
df.mepkviq$kviq.intensity = 0
for (pid in unique(df.mepkviq$pid)) {
  # get kviq values
  pid.kviq.ima = unique(df.eegbci`KVIQ Imagery`[df.eegbci$PID == pid])
  pid.kviq.int = unique(df.eegbci`KVIQ Intensity`[df.eegbci$PID == pid])
  pid.kviq.tot = unique(df.eegbci`KVIQ Total`[df.eegbci$PID == pid])

  df.mepkviq$kviq.imagery[df.mepkviq$pid == pid] = pid.kviq.ima
  df.mepkviq$kviq.intensity[df.mepkviq$pid == pid] = pid.kviq.int
  df.mepkviq$kviq.total[df.mepkviq$pid == pid] = pid.kviq.tot
}

# checking out kviq
base.model = lmer(Dynamic.Standardised.MEP ~ conditions * mep.nr.continuous + (1|pid), data = df.mepkviq)
base.ztrans = lmer(Dynamic.Standardised.MEP ~ conditions * mep.nr.continuous + ztrans.rms + (1|pid), data = df.mepkviq)
base.kviq = lmer(Dynamic.Standardised.MEP ~ conditions * mep.nr.continuous + kviq.total + (1|pid), data = df.mepkviq)
kviq.ztrans = lmer(Dynamic.Standardised.MEP ~ conditions * mep.nr.continuous + kviq.total + ztrans.rms + (1|pid), data = df.mepkviq)
fullint.kviq = lmer(Dynamic.Standardised.MEP ~ conditions * mep.nr.continuous * kviq.total + (1|pid), data = df.mepkviq)
fullint.ztrans = lmer(Dynamic.Standardised.MEP ~ conditions * mep.nr.continuous * ztrans.rms + (1|pid), data = df.mepkviq)
fullint.model = lmer(Dynamic.Standardised.MEP ~ conditions * mep.nr.continuous * ztrans.rms * kviq.total + (1|pid), data = df.mepkviq)
summary(base.model)
summary(base.ztrans)
summary(base.kviq)
summary(kviq.ztrans)
summary(fullint.kviq)
format_coeff_table_apa(fullint.kviq,
  save.path.name = "../Processed Data/MEP Data/Model Data/Mkviq/kviq-coeff",
  replace.variables = c(),
  replacement.variables = c())
summary(fullint.ztrans)

summary(fullint.model)
anova(base.model, base.ztrans, base.kviq, kviq.ztrans, fullint.kviq, fullint.ztrans, fullint.model)

model.mep.kviq.1 = lmer(Dynamic.Standardised.MEP ~ conditions * mep.nr.continuous * kviq.total + (1|pid), data = df.mepkviq)
model.mep.kviq.2 = lmer(Dynamic.Standardised.MEP ~ conditions * mep.nr.continuous * kviq.imagery + (1|pid), data = df.mepkviq)
model.mep.kviq.3 = lmer(Dynamic.Standardised.MEP ~ conditions * mep.nr.continuous * kviq.intensity + (1|pid), data = df.mepkviq)
summary(model.mep.kviq.1)
summary(model.mep.kviq.2)
summary(model.mep.kviq.3)
anova(model.mep.kviq.1, model.mep.kviq.2, model.mep.kviq.3)

model.rawmep.kviq.1 = lmer(actual_MEP ~ conditions * mep.nr.continuous * kviq.total + (1|pid), data = df.mepkviq)
model.rawmep.kviq.2 = lmer(actual_MEP ~ conditions * mep.nr.continuous * kviq.imagery + (1|pid), data = df.mepkviq)
model.rawmep.kviq.3 = lmer(actual_MEP ~ conditions * mep.nr.continuous * kviq.intensity + (1|pid), data = df.mepkviq)
summary(model.rawmep.kviq.1)

```



```

summary(model.rawmep.kviq.2)
summary(model.rawmep.kviq.3)
anova(model.rawmep.kviq.1, model.rawmep.kviq.2, model.rawmep.kviq.3)

# check out kviq & brms
model.mepbrms.kviq.1 = lmer(actual_MEP ~ conditions * mep.nr.continuous * kviq.total *
ztrans.rms + (1|pid), data = df.mepkviq)
model.mepbrms.kviq.2 = lmer(actual_MEP ~ conditions * mep.nr.continuous * kviq.imagery *
ztrans.rms + (1|pid), data = df.mepkviq)
summary(model.mepbrms.kviq.1)
summary(model.mepbrms.kviq.2)
anova(model.mepbrms.kviq.1, model.mepbrms.kviq.2)

# Publication Models -----
## MEP -----
# Model nr 1
save.path = "./Processed Data/MEP Data/Publication/Model 1/Basic Model"
pub.mod.1 = Model_me_this(model.data = df.updown,
                          model.formula = "actual_MEP ~ conditions * mep.nr.train +
(1|pid)",
                          save_path = save.path,
                          plot.model = FALSE)
pub.coeff.1 = format_coeff_table_apa(model = pub.mod.1$LM$Model,
                                     save.path.name = save.path,
                                     replace.variables = c("conditionsup", "mep.nr.train"),
                                     replacement.variables = c("Condition (Up)", "Time"))

# Model nr 2
save.path = "./Processed Data/MEP Data/Publication/Model 2/Session Model"
pub.mod.2 = Model_me_this(model.data = df.updown,
                          model.formula = "actual_MEP ~ conditions * mep.nr.train + sess +
eeg.condition + (1|pid)",
                          save_path = save.path,
                          plot.model = FALSE)
pub.coeff.2 = format_coeff_table_apa(model = pub.mod.2$LM$Model,
                                     save.path.name = save.path,
                                     replace.variables = c("conditionsup", "mep.nr.train",
"sess", "eeg.condition"),
                                     replacement.variables = c("Condition (Up)", "Time",
"Second Session", "EEG Session"))

# chi square test
comp.m1.m2 = anova(pub.mod.1$LM$Model, pub.mod.2$LM$Model)
write(capture.output(comp.m1.m2), file = "./Processed Data/MEP
Data/Publication/m1vm2comp.txt")

# Model nr 3
save.path = "./Processed Data/MEP Data/Publication/Model 3/BrMS Model"
pub.mod.3 = Model_me_this(model.data = df.updown,
                          model.formula = "actual_MEP ~ conditions * mep.nr.train +
ztrans.rms + ztrans.rms:conditions + (1|pid)",
                          save_path = save.path,
                          plot.model = FALSE)
pub.coeff.3 = format_coeff_table_apa(model = pub.mod.3$LM$Model,
                                     save.path.name = save.path,
                                     replace.variables = c("conditionsup", "mep.nr.train",
"ztrans.rms"),
                                     replacement.variables = c("Condition (Up)", "Time",
"BrMS"))
comp.m2.m3 = anova(pub.mod.2$LM$Model, pub.mod.3$LM$Model)
library(performance)
test = compare_performance(pub.mod.2$LM$Model, pub.mod.3$LM$Model, rank = TRUE)

# Model nr 4

```

```

save.path = "../Processed Data/MEP Data/Publication/Model 4/pred BrMS Model"
pub.mod.4 = Model_me_this(model.data = df.updown,
                          model.formula = "ztrans.rms ~ conditions * mep.nr.train +
actual_MEP + (1|pid)",
                          save_path = save.path,
                          plot.model = FALSE)
pub.coeff.4 = format_coeff_table_apa(model = pub.mod.4$LM$Model,
                                     save.path.name = save.path,
                                     replace.variables = c("conditionsup", "mep.nr.train",
"actual_MEP"),
                                     replacement.variables = c("Condition (Up)", "Time",
"MEP Amplitude"))
pub.coeff.4

# Model nr 5
save.path = "../Processed Data/MEP Data/Publication/Model 5/dyn BrMS Model"
pub.mod.5 = Model_me_this(model.data = df.updown,
                          model.formula = "Dynamic.Standardised.MEP ~ conditions *
mep.nr.train + ztrans.rms + ztrans.rms:conditions + (1|pid)",
                          save_path = save.path,
                          plot.model = FALSE)
pub.coeff.5 = format_coeff_table_apa(model = pub.mod.5$LM$Model,
                                     save.path.name = save.path,
                                     replace.variables = c("conditionsup", "mep.nr.train",
"ztrans.rms"),
                                     replacement.variables = c("Condition (Up)", "Time",
"BrMS"))
pub.coeff.5

# Model nr 6
save.path = "../Processed Data/MEP Data/Publication/Model 6/Kviq Model"
pub.mod.6 = Model_me_this(model.data = df.mepkviq,
                          model.formula = "actual_MEP ~ conditions * mep.nr.train *
kviq.total + eeg.condition + sess + (1|pid)",
                          save_path = save.path,
                          plot.model = FALSE)
pub.coeff.6 = format_coeff_table_apa(model = pub.mod.6$LM$Model,
                                     save.path.name = save.path,
                                     replace.variables = c("conditionsup", "mep.nr.train",
"kviq.total"),
                                     replacement.variables = c("Condition (Up)", "Time",
"KVIQ"))
pub.coeff.6

# Best fullest model?
save.path = "../Processed Data/MEP Data/Publication/Model 7/fullint Model"
pub.mod.fullint = Model_me_this(model.data = df.mepkviq,
                                model.formula = "actual_MEP ~ conditions * mep.nr.train *
kviq.total * ztrans.rms + eeg.condition + sess + (1|pid)",
                                save_path = save.path,
                                plot.model = FALSE)
pub.coeff.fullint = format_coeff_table_apa(model = pub.mod.fullint$LM$Model,
                                           save.path.name = save.path,
                                           replace.variables = c("conditionsup",
"mep.nr.train", "kviq.total",
"ztrans.rms"),
                                           "eeg.condition", "sess",
                                           replacement.variables = c("Condition (Up)",
"Time", "KVIQ",
"EEG Session", "Second
Session", "BrMS"))

save.path = "../Processed Data/MEP Data/Publication/Model 7/ztrans Model"
pub.mod.ztrans = Model_me_this(model.data = df.mepkviq,
                                model.formula = "actual_MEP ~ conditions * mep.nr.train *
ztrans.rms + eeg.condition + sess + (1|pid)",
                                save_path = save.path,
                                plot.model = FALSE)

```

```

pub.coeff.ztrans = format_coeff_table_apa(model = pub.mod.ztrans$LM$Model,
                                          save.path.name = save.path,
                                          replace.variables = c("conditionsup",
                                                                "eeg.condition", "sess",
                                                                "ztrans.rms"),
                                          replacement.variables = c("Condition (Up)", "Time",
                                                                    "EEG Session", "Second
                                                                    Session", "BrMS"))

save.path = "./Processed Data/MEP Data/Publication/Model 7/kviq Model"
pub.mod.kviq = Model_me_this(model.data = df.mepkviq,
                             model.formula = "actual_MEP ~ conditions * mep.nr.train *
kviq.total + eeg.condition + sess + (1|pid)",
                             save_path = save.path,
                             plot.model = FALSE)
pub.coeff.kviq = format_coeff_table_apa(model = pub.mod.kviq$LM$Model,
                                          save.path.name = save.path,
                                          replace.variables = c("conditionsup",
                                                                "eeg.condition", "sess"),
                                          replacement.variables = c("Condition (Up)",
                                                                    "EEG Session", "Second
                                                                    Session"))
pub.coeff.kviq
pub.coeff.ztrans
pub.coeff.fullint
anova(pub.mod.ztrans$LM$Model, pub.mod.kviq$LM$Model, pub.mod.fullint$LM$Model)

# CA -----
save.path = "./Processed Data/CA Data/Publication/Model 1/Base Model"
pub.ca.1 = Model_me_this(model.data = df.eegbci,
                         model.formula = "CA ~ Group * Continuous.Block + (1|PID)",
                         save_path = save.path,
                         plot.model = FALSE)
pub.ca.coeff.1 = format_coeff_table_apa(model = pub.ca.1$LM$Model,
                                          save.path.name = save.path,
                                          replace.variables = c("GroupTMS",
                                                                "Continuous.Block"),
                                          replacement.variables = c("Group (TMS)", "Time"))
pub.ca.coeff.1

save.path = "./Processed Data/CA Data/Publication/Model 2/KSS Model"
load(file = "./Processed Data/CA Data/Model/mKSS/mKSS-KSS-model.rda")
check_lm_assumptions(Linear_Model,
                      save_path = save.path)

save.path = "./Processed Data/CA Data/Publication/Model 3/KSS Model"
load(file = "./Processed Data/CA Data/Model/mKSS/mKSS-KSSGroupIntComp-model.rda")
check_lm_assumptions(Linear_Model,
                      save_path = save.path)

```

10.4.14 MEP Preprocessing (R)

This data takes most of the preprocessed MEP (in Matlab format) and makes a dataframe ready for building model and testing hypotheses.

```
#clear all
rm(list = ls())

# Libraries and read in Data -----
library(ggplot2)
library(readr)
library(plyr)
library(dplyr)
library(PupillometryR)
source("https://raw.githubusercontent.com/datavizpyr/data/master/half_flat_violinplot.R")
library(tidyverse)
library(plotrix)
library(readxl)

# Go into Folder
work_dir = ""
setwd(work_dir)

# read in data
data.dir = "../"
# get all files
all_files = list.files(path = data.dir,
                      recursive = TRUE,
                      include.dirs = FALSE,
                      pattern = "_MEPs.txt")

results = data.frame()
print(all_files)

# Load all files
for (i in 1:length(all_files)) {
  current_table = read.delim(paste(data.dir, "/", all_files[i], sep = ""))
  current_table = current_table[,c(1,4:22)]
  # Find PID
  pid = c(rep(strsplit(strsplit(all_files[i], split = "/")[[1]][1],
                          split = "nf")[[1]][2],
              dim(current_table)[1]))
  # Find Sess nr
  sess = c(rep(strsplit(strsplit(all_files[i], split = "/")[[1]][2],
                          split = "sess")[[1]][2],
              dim(current_table)[1]))

  current_table = cbind(pid, sess, current_table)

  results = rbind.data.frame(results, current_table)
}

## Data Management -----
# Manage File names
filenames = character()
# get condition of meps
for (n in 1:dim(results)[1]) {
  current_filename = gsub("\\\\", "/", results$Filename[n])
  filenames[n] = strsplit(current_filename,
                          split = "/")[[1]][length(strsplit(current_filename, split =
"/")[[1]])]

  # if filename is too short, add PID and Sess to homogenise
  # if this does not work, try to "find(name, "TMS-time")" might be more elegant
}
```

```

if (nchar(filenames[n]) < 30){
  # add PID and Sess
  old.split = strsplit(filenames[n], '-')
  new.name = paste('sub', results$pid[n], old.split[[1]][1],
                  'sess', results$sess[n],
                  paste(old.split[[1]][2:length(old.split[[1])]), sep = "", collapse =
"-"),
                  sep = '-')
  filenames[n] = new.name
  remove(new.name)
}
}

results = cbind.data.frame(filenames, results)

# get out blocks & conditions
Condition.Names = c("pre", "post", "up", "down")
conditions = character()
eeg.condition = logical()
block_nr = character()
# get condition of meps
for (n in 1:dim(results)[1]) {
  name.split = strsplit(results$filenames[n],
                        split = "-")[[1]]
  name.ind = Condition.Names %in% name.split
  eeg.condition[n] = as.logical(length(grep('eeg', name.split)))
  conditions[n] = Condition.Names[name.ind]
  # I don't think we collect pre-eeg with the scripts for now. so let take it out for now
  if (conditions[n] == "pre") {
    if (eeg.condition[n]){
      block_nr[n] = "pre-eeg"
    } else {
      block_nr[n] = "pre"
    }
  } else if (conditions[n] == "post") {
    if (eeg.condition[n]){
      block_nr[n] = "post-eeg"
    } else {
      block_nr[n] = "post"
    }
  } else {
    block_nr[n] = strsplit(strsplit(results$filenames[n], split = "-")[[1]][5],
                          split = "block")[[1]][2]
  }
}
results = cbind.data.frame(conditions, block_nr, eeg.condition, results)

# add results mep nr
mep.nr = numeric()
for (i in 1:dim(results)[1]) {
  mep.nr.string = strsplit(results$filenames[i], split = "-")[[1]]
  mep.nr.ind = grep('mep', mep.nr.string)
  mep.nr[i] = parse_number(mep.nr.string[mep.nr.ind])
}

results$mep.nr = mep.nr

## Add continuous number of meps
mep.nr.continuous = numeric()
for (i in 1:dim(results)[1]) {
  # determine session, session 2 adds 120 meps
  if (results$sess[i] == "1"){
    mepsess = 0
  } else {
    mepsess = 120
  }
}

# determine block. every block adds 20 meps, pre adds 0, post adds 100

```

```

if (grepl('pre', results$block_nr[i])){
  mepblock = 0
  if (results$eeg.condition[i]){
    mepblock = 120
  }
} else if (grepl('post', results$block_nr[i])) {
  mepblock = 100
  if (results$eeg.condition[i]){
    mepblock = 180
  }
} else {
  mepblock = as.numeric(results$block_nr[i]) * 20
  # if it's block 5 or 6 add 20 for pre block
  if (mepblock == 100){
    mepblock = 140
  } else if (mepblock == 120){
    mepblock = 160
  }
}
mep.nr.continuous[i] = mepsess + mepblock + results$mep.nr[i]
}

results$mep.nr.continuous = mep.nr.continuous

## add number depending on condition so we want s1d1 = 1-20, s1d2 = 21-40 etc
# create train nr vec
results$mep.nr.train = NA
# order dataframe so mep.nr.cont is ascending
results = results[order(results$pid, results$sess, results$mep.nr.continuous), ]
for (i in 1:length(unique(results$pid))){
  current.pid = unique(results$pid)[i]
  # Get indices of current pid down sessions
  current.down.ind = which(results$conditions == "down" & results$pid == current.pid)
  # Get indices of current pid up sessions
  current.up.ind = which(results$conditions == "up" & results$pid == current.pid)

  results$mep.nr.train[current.down.ind] = 1:length(current.down.ind)
  results$mep.nr.train[current.up.ind] = 1:length(current.up.ind)
}

# Add "visual reject" label -----
# read in visual rejects table
vis.rej = read_excel(path = "../MEP_mat_files/vis_abnorm.xlsx")
vis.rej$UID = interaction(parse_number(vis.rej$Nf),
  parse_number(vis.rej$Session),
  vis.rej$Condition,
  parse_number(vis.rej$Block),
  parse_number(vis.rej$MEP),
  vis.rej$EEG)

results <- results %>%
  mutate(uidblock = recode(block_nr, "pre" = "0", "post" = "0",
    "pre-eeg" = "0", "post-eeg" = "0"))

results$UID = interaction(results$pid,
  results$sess,
  results$conditions,
  results$uidblock,
  results$mep.nr,
  as.numeric(results$eeg.condition))

results$Visual.Reject = results$UID %in% vis.rej$UID

# Data Manipulations -----
# cut into bins

```

```

results$Thresh_rFDI = cut(results$pre_rms_Ch2, breaks=c(-Inf, 0.01, Inf), labels =
c("Below", "Above"))
results$Thresh_rADM = cut(results$pre_rms_Ch3, breaks=c(-Inf, 0.01, Inf), labels =
c("Below", "Above"))
results$Thresh_lFDI = cut(results$pre_rms_Ch4, breaks=c(-Inf, 0.01, Inf), labels =
c("Below", "Above"))
results$Thresh_lADM = cut(results$pre_rms_Ch5, breaks=c(-Inf, 0.01, Inf), labels =
c("Below", "Above"))

# actual mep, divided by 2
results$actual_MEP = (results$Peak_Peak_Ch2 + results$Peak_Peak_Ch3)/2
results$actual_rMS_right = (results$pre_rms_Ch2 + results$pre_rms_Ch3)/2

# Standardisation
results$Standardised.Threshold = NA
# read in threshold used per session:
for (i in 1:length(unique(results$pid))) {
  current.pid = unique(results$pid)[i]
  # read in threshold file
  sess1.txt = read_file(file = paste('../.../2 Stage BCI Sync Folder/Complete Data/nf',
current.pid, '/sess1/pre-mep/Profile-pre.txt',
sep = ''))

  # split text to get number
  sess1.txt.split = strsplit(sess1.txt, split = "\n")
  sess1.txt.split = strsplit(sess1.txt.split[[1]][8], split = "was ")
  sess1.thresh = as.numeric(sess1.txt.split[[1]][2])
  # Adjust for Gain Manually here
  results$Standardised.Threshold[results$pid == current.pid & results$sess == 1] =
sess1.thresh/2

  # get sess 2 thresh
  # read in threshold file
  sess2.txt = read_file(file = paste('../.../2 Stage BCI Sync Folder/Complete Data/nf',
current.pid, '/sess2/pre-mep/Profile-pre.txt',
sep = ''))

  # split text to get number
  sess2.txt.split = strsplit(sess2.txt, split = "\n")
  sess2.txt.split = strsplit(sess2.txt.split[[1]][8], split = "was ")
  sess2.thresh = as.numeric(sess2.txt.split[[1]][2])
  # Adjust for Gain Manually here
  results$Standardised.Threshold[results$pid == current.pid & results$sess == 2] =
sess2.thresh/2

  # get sess 2 eeg thresh
  # read in threshold file
  sess2.eeg.txt = read_file(file = paste('../.../2 Stage BCI Sync Folder/Complete Data/nf',
current.pid, '/sess2/pre-mep/Profile-pre-eeg.txt',
sep = ''))

  # split text to get number
  sess2.eeg.txt.split = strsplit(sess2.eeg.txt, split = "\n")
  sess2.eeg.txt.split = strsplit(sess2.eeg.txt.split[[1]][8], split = "was ")
  sess2.eeg.thresh = as.numeric(sess2.eeg.txt.split[[1]][2])
  # Adjust for Gain Manually here
  results$Standardised.Threshold[results$pid == current.pid & results$sess == 2 &
results$eeg.condition == TRUE] = sess2.eeg.thresh/2
}

# Adjust MEP to standardisation
results$Standardised.MEP = NA
for (i in 1:dim(results)[1]) {
  results$Standardised.MEP[i] = results$actual_MEP[i] - results$Standardised.Threshold[i]
}

# preproc Standardisation
# use all collected meps during blocks and use average as threshold
results$Preproc.Standardised.Threshold = NA
for (i in 1:length(unique(results$pid))) {
  # get threshold sess1

```

```

    current.pid = unique(results$pid)[i]
    current.pre.threshold.sess1 = mean(results$actual_MEP[results$block_nr == "pre" &
results$pid == current.pid &
                                results$sess == 1])
    results$Preproc.Standardised.Threshold[results$pid == current.pid & results$sess == 1] =
current.pre.threshold.sess1

    # get threshold sess2
    current.pre.threshold.sess2 = mean(results$actual_MEP[results$block_nr == "pre" &
results$pid == current.pid &
                                results$sess == 2])
    results$Preproc.Standardised.Threshold[results$pid == current.pid & results$sess == 2 &
results$eeg.condition == FALSE] = current.pre.threshold.sess2

    # get threshold sess2 eeg
    current.pre.threshold.sess2.eeg = mean(results$actual_MEP[results$block_nr == "pre-eeg" &
results$pid == current.pid &
                                results$sess == 2])
    results$Preproc.Standardised.Threshold[results$pid == current.pid & results$sess == 2 &
results$eeg.condition == TRUE] = current.pre.threshold.sess2.eeg
}
# Adjust MEP to preproc standardisation
results$Preproc.Standardised.MEP = NA
for (i in 1:dim(results)[1]) {
    results$Preproc.Standardised.MEP[i] = results$actual_MEP[i] -
results$Preproc.Standardised.Threshold[i]
}

# Dynamic Standardisation
# use all collected meps during blocks and use average as threshold
results$Dynamic.Threshold = NA
for (i in 1:length(unique(results$pid))) {
    # get threshold sess1
    current.pid = unique(results$pid)[i]
    current.dynamic.threshold.sess1 = mean(results$actual_MEP[results$block_nr %in% c("1",
"2", "3", "4") & results$pid == current.pid &
                                results$sess == 1])
    results$Dynamic.Threshold[results$pid == current.pid & results$sess == 1] =
current.dynamic.threshold.sess1

    # get threshold sess2
    current.dynamic.threshold.sess2 = mean(results$actual_MEP[results$block_nr %in% c("1",
"2", "3", "4") & results$pid == current.pid &
                                results$sess == 2])
    results$Dynamic.Threshold[results$pid == current.pid & results$sess == 2 &
results$eeg.condition == FALSE] = current.dynamic.threshold.sess2

    # get threshold sess2 eeg
    current.dynamic.threshold.sess2.eeg = mean(results$actual_MEP[results$block_nr %in%
c("5", "6") & results$pid == current.pid &
                                results$sess == 2])
    results$Dynamic.Threshold[results$pid == current.pid & results$sess == 2 &
results$eeg.condition == TRUE] = current.dynamic.threshold.sess2.eeg
}
# Adjust MEP to preproc standardisation
results$Dynamic.Standardised.MEP = NA
for (i in 1:dim(results)[1]) {
    results$Dynamic.Standardised.MEP[i] = results$actual_MEP[i] -
results$Dynamic.Threshold[i]
}

# z.Transformation Standardisation
# use all collected meps during blocks and use average as threshold
results$z.Transformation.Mean = NA
results$z.Transformation.SD = NA
for (i in 1:length(unique(results$pid))) {
    # get threshold sess1
    current.pid = unique(results$pid)[i]

```



```

    current.z.Transformation.sess1.mean = mean(results$actual_MEP[results$block_nr %in%
c("1", "2", "3", "4") & results$pid == current.pid &
                                results$sess == 1])
    current.z.Transformation.sess1.sd = sd(results$actual_MEP[results$block_nr %in% c("1",
"2", "3", "4") & results$pid == current.pid &
                                results$sess == 1])
    results$z.Transformation.Mean[results$pid == current.pid & results$sess == 1] =
current.z.Transformation.sess1.mean
    results$z.Transformation.SD[results$pid == current.pid & results$sess == 1] =
current.z.Transformation.sess1.sd

    # get threshold sess2
    current.z.Transformation.sess2.mean = mean(results$actual_MEP[results$block_nr %in%
c("1", "2", "3", "4") & results$pid == current.pid &
                                results$sess == 2])
    current.z.Transformation.sess2.sd = sd(results$actual_MEP[results$block_nr %in% c("1",
"2", "3", "4") & results$pid == current.pid &
                                results$sess == 2])
    results$z.Transformation.Mean[results$pid == current.pid & results$sess == 2 &
results$eeg.condition == FALSE] = current.z.Transformation.sess2.mean
    results$z.Transformation.SD[results$pid == current.pid & results$sess == 2 &
results$eeg.condition == FALSE] = current.z.Transformation.sess2.sd

    # get threshold sess2 eeg
    current.z.Transformation.sess2.mean.eeg = mean(results$actual_MEP[results$block_nr %in%
c("5", "6") & results$pid == current.pid &
                                results$sess == 2])
    current.z.Transformation.sess2.sd.eeg = mean(results$actual_MEP[results$block_nr %in%
c("5", "6") & results$pid == current.pid &
                                results$sess == 2])
    results$z.Transformation.Mean[results$pid == current.pid & results$sess == 2 &
results$eeg.condition == TRUE] = current.z.Transformation.sess2.mean.eeg
    results$z.Transformation.SD[results$pid == current.pid & results$sess == 2 &
results$eeg.condition == TRUE] = current.z.Transformation.sess2.sd.eeg
}
# Adjust MEP to preproc standardisation
results$z.Transformation.Standardised.MEP = NA
for (i in 1:dim(results)[1]) {
    results$z.Transformation.Standardised.MEP[i] = (results$actual_MEP[i] -
results$z.Transformation.Mean[i])/results$z.Transformation.SD[i]
}

# create artificial S3 for eeg
results$plot.sessions = results$sess
results$plot.sessions[results$eeg.condition] = "3"

# create artificial blocks for training all blocks
results$plot.blocks = NA
for (i in 1:dim(results)[1]) {
    curr.sess = results$plot.sessions[i]
    curr.pid = results$pid[i]
    curr.cond = results$conditions[i]
    curr.blocks = unique(results$block_nr[results$sess == curr.sess &
results$pid == curr.pid])

    if (curr.cond == "pre"){
        results$plot.blocks[i] = NA
    } else if (curr.cond == "post"){
        results$plot.blocks[i] = NA
    } else if (curr.cond == "down"){
        # Get block numbers and sort them numerically
        curr.cond.blocks = sort(as.numeric(
            unique(results$block_nr[results$plot.sessions == curr.sess &
results$pid == curr.pid &
results$conditions == "down"])))
        # if in the lower number and sess1 = 1, higher number = 2. Sess2 = 3, 4. Sess3 = 5,6
        if (curr.sess == 1){
            # if current block is same as first item in sorted list of blocks for this
            session/participant

```

```

    if ((curr.cond.blocks[1] == results$block_nr[i])){
      results$plot.blocks[i] = 1
      # if it is equal the second item, blocknr 2
    } else if ((curr.cond.blocks[2] == results$block_nr[i])){
      results$plot.blocks[i] = 2
    }
  } # if session 2...
  if (curr.sess == 2){# if current block is same as first item in sorted list of blocks
for this session/participant
    if ((curr.cond.blocks[1] == results$block_nr[i])){
      results$plot.blocks[i] = 3
      # if it is equal the second item, blocknr 2
    } else if ((curr.cond.blocks[2] == results$block_nr[i])){
      results$plot.blocks[i] = 4
    }
  }
  if (curr.sess == 3){# if current block is same as first item in sorted list of blocks
for this session/participant
    if ((curr.cond.blocks[1] == results$block_nr[i])){
      results$plot.blocks[i] = 5
      # if it is equal the second item, blocknr 2
    } else if ((curr.cond.blocks[2] == results$block_nr[i])){
      results$plot.blocks[i] = 6
    }
  } # now if up condition...
} else if (curr.cond == "up"){
  # Get block numbers and sort them numerically
  curr.cond.blocks = sort(as.numeric(
    unique(results$block_nr[results$plot.sessions == curr.sess &
      results$pid == curr.pid &
      results$conditions == "up"])))
  # if in the lower number and sess1 = 1, higher number = 2. Sess2 = 3, 4. Sess3 = 5,6
  if (curr.sess == 1){
    # if current block is same as first item in sorted list of blocks for this
session/participant
    if ((curr.cond.blocks[1] == results$block_nr[i])){
      results$plot.blocks[i] = 1
      # if it is equal the second item, blocknr 2
    } else if ((curr.cond.blocks[2] == results$block_nr[i])){
      results$plot.blocks[i] = 2
    }
  } # if session 2...
  if (curr.sess == 2){# if current block is same as first item in sorted list of blocks
for this session/participant
    if ((curr.cond.blocks[1] == results$block_nr[i])){
      results$plot.blocks[i] = 3
      # if it is equal the second item, blocknr 2
    } else if ((curr.cond.blocks[2] == results$block_nr[i])){
      results$plot.blocks[i] = 4
    }
  }
  if (curr.sess == 3){# if current block is same as first item in sorted list of blocks
for this session/participant
    if ((curr.cond.blocks[1] == results$block_nr[i])){
      results$plot.blocks[i] = 5
      # if it is equal the second item, blocknr 2
    } else if ((curr.cond.blocks[2] == results$block_nr[i])){
      results$plot.blocks[i] = 6
    }
  }
}
}
}
remove(curr.blocks,
  curr.cond,
  curr.cond.blocks,
  curr.pid,
  curr.sess)

```

```

remove(sess1.thresh,
       sess1.txt,
       sess1.txt.split,
       sess2.eeg.thresh,
       sess2.eeg.txt,
       sess2.eeg.txt.split,
       sess2.thresh,
       sess2.txt,
       old.split,
       sess,
       pid,
       name.split,
       name.ind,
       n,
       mepsess,
       mepblock,
       mep.nr,
       mep.nr.continuous,
       mep.nr.ind,
       mep.nr.string,
       i,
       filenames,
       eeg.condition,
       current_filename,
       current_table,
       current.down.ind,
       current.dynamic.threshold.sess1,
       current.dynamic.threshold.sess2,
       current.dynamic.threshold.sess2.eeg,
       current.pid,
       current.pre.threshold.sess1,
       current.pre.threshold.sess2,
       current.pre.threshold.sess2.eeg,
       current.up.ind,
       current.z.Transformation.sess1.mean,
       current.z.Transformation.sess1.sd,
       current.z.Transformation.sess2.mean,
       current.z.Transformation.sess2.mean.eeg,
       current.z.Transformation.sess2.sd,
       current.z.Transformation.sess2.sd.eeg,
       conditions,
       block_nr,
       sess2.txt.split)

# Success Metrics -----

# Standardised Success
UpSucc = ((results$Standardised.MEP > 0) & (results$conditions == "up"))
DnSucc = ((results$Standardised.MEP < 0) & (results$conditions == "down"))
results$Standardised.Success = UpSucc | DnSucc
results$Standardised.Success[!results$conditions %in% c("up", "down")] = NA

plot(results$Standardised.MEP[UpSucc])
plot(results$Standardised.MEP[DnSucc])

# Standardised Success
UpSucc = ((results$Preproc.Standardised.MEP > 0) & (results$conditions == "up"))
DnSucc = ((results$Preproc.Standardised.MEP < 0) & (results$conditions == "down"))
results$Preproc.Standardised.Success = UpSucc | DnSucc
results$Preproc.Standardised.Success[!results$conditions %in% c("up", "down")] = NA

plot(results$Preproc.Standardised.MEP[UpSucc])
plot(results$Preproc.Standardised.MEP[DnSucc])

# Dynamic Success
UpSucc = ((results$Dynamic.Standardised.MEP > 0) & (results$conditions == "up"))

```

```

DnSucc = ((results$Dynamic.Standardised.MEP < 0) & (results$conditions == "down"))
results$Dynamic.Standardised.Success = UpSucc | DnSucc
results$Dynamic.Standardised.Success[!results$conditions %in% c("up", "down")] = NA

plot(results$Dynamic.Standardised.MEP[UpSucc])
plot(results$Dynamic.Standardised.MEP[DnSucc])

# zTransformed Success
UpSucc = ((results$z.Transformation.Standardised.MEP > 0) & (results$conditions == "up"))
DnSucc = ((results$z.Transformation.Standardised.MEP < 0) & (results$conditions == "down"))
results$zTransformed.Standardised.Success = UpSucc | DnSucc
results$zTransformed.Standardised.Success[!results$conditions %in% c("up", "down")] = NA

plot(results$z.Transformation.Standardised.MEP[UpSucc])
plot(results$z.Transformation.Standardised.MEP[DnSucc])

# Lastly check for duplicates
# in UID
results[duplicated(results$UID),]

# in Filenames
results[duplicated(results$Filename),]

# Save DF -----

save(results, file = './Processed Data/MEP Data/MEP Data.Rda')
#
# #
# boxplot(results$actual_rMS_right ~ results$pid)
# boxplot(results$Dynamic.Standardised.MEP ~ interaction(results$pid, results$conditions))

# ## Plot functions -----
# dataframe = results
# participant = "1"
# y.axis = "Dynamic.Standardised.MEP"
# x.axis = "mep.nr.train"
# # plot order actually need to label pre from pre1 to pre3
# dataframe$plot.Order = interaction(dataframe$plot.blocks, dataframe$conditions)
# dataframe$plot.Order[dataframe$conditions == "pre"] =
interaction(dataframe$plot.sessions[dataframe$conditions == "pre"], "pre")
# dataframe$plot.Order = factor(dataframe$plot.Order, levels = c("1.pre", "1.down", "1.up",
"2.down", "2.up",
"2.pre", "3.down", "3.up",
"4.down", "4.up",
"3.pre", "5.down",
"5.up"))
#
# pre.mean.1 = mean(dataframe$actual_MEP[dataframe$plot.Order == "1.pre"], na.rm = TRUE)
# pre.mean.2 = mean(dataframe$actual_MEP[dataframe$plot.Order == "2.pre"], na.rm = TRUE)
# pre.mean.3 = mean(dataframe$actual_MEP[dataframe$plot.Order == "3.pre"], na.rm = TRUE)
# up.mean.1 = mean(dataframe$actual_MEP[dataframe$plot.Order == "1.up"], na.rm = TRUE)
# up.mean.2 = mean(dataframe$actual_MEP[dataframe$plot.Order == "2.up"], na.rm = TRUE)
# up.mean.3 = mean(dataframe$actual_MEP[dataframe$plot.Order == "3.up"], na.rm = TRUE)
# up.mean.4 = mean(dataframe$actual_MEP[dataframe$plot.Order == "4.up"], na.rm = TRUE)
# up.mean.5 = mean(dataframe$actual_MEP[dataframe$plot.Order == "5.up"], na.rm = TRUE)
# down.mean.1 = mean(dataframe$actual_MEP[dataframe$plot.Order == "1.down"], na.rm = TRUE)
# down.mean.2 = mean(dataframe$actual_MEP[dataframe$plot.Order == "2.down"], na.rm = TRUE)
# down.mean.3 = mean(dataframe$actual_MEP[dataframe$plot.Order == "3.down"], na.rm = TRUE)
# down.mean.4 = mean(dataframe$actual_MEP[dataframe$plot.Order == "4.down"], na.rm = TRUE)
# down.mean.5 = mean(dataframe$actual_MEP[dataframe$plot.Order == "5.down"], na.rm = TRUE)
# MEP.Mean = c(pre.mean.1, pre.mean.2, pre.mean.3, up.mean.1, up.mean.2, up.mean.3,
# up.mean.4, up.mean.5, down.mean.1, down.mean.2, down.mean.3, down.mean.4,
# down.mean.5)
#
# pre.stderr.1 = std.error(dataframe$actual_MEP[dataframe$plot.Order == "1.pre"], na.rm =
TRUE)
# pre.stderr.2 = std.error(dataframe$actual_MEP[dataframe$plot.Order == "2.pre"], na.rm =
TRUE)

```

```

# pre.stdev.3 = std.error(dataframe$actual_MEP[dataframe$plot.Order == "3.pre"], na.rm =
TRUE)
# up.stdev.1 = std.error(dataframe$actual_MEP[dataframe$plot.Order == "1.up"], na.rm =
TRUE)
# up.stdev.2 = std.error(dataframe$actual_MEP[dataframe$plot.Order == "2.up"], na.rm =
TRUE)
# up.stdev.3 = std.error(dataframe$actual_MEP[dataframe$plot.Order == "3.up"], na.rm =
TRUE)
# up.stdev.4 = std.error(dataframe$actual_MEP[dataframe$plot.Order == "4.up"], na.rm =
TRUE)
# up.stdev.5 = std.error(dataframe$actual_MEP[dataframe$plot.Order == "5.up"], na.rm =
TRUE)
# down.stdev.1 = std.error(dataframe$actual_MEP[dataframe$plot.Order == "1.down"], na.rm =
TRUE)
# down.stdev.2 = std.error(dataframe$actual_MEP[dataframe$plot.Order == "2.down"], na.rm =
TRUE)
# down.stdev.3 = std.error(dataframe$actual_MEP[dataframe$plot.Order == "3.down"], na.rm =
TRUE)
# down.stdev.4 = std.error(dataframe$actual_MEP[dataframe$plot.Order == "4.down"], na.rm =
TRUE)
# down.stdev.5 = std.error(dataframe$actual_MEP[dataframe$plot.Order == "5.down"], na.rm =
TRUE)
# MEP.stdev = c(pre.stdev.1, pre.stdev.2, pre.stdev.3, up.stdev.1, up.stdev.2,
#               up.stdev.3, up.stdev.4, up.stdev.5, down.stdev.1, down.stdev.2,
#               down.stdev.3, down.stdev.4, down.stdev.5)
# MEP.df = data.frame(MEP.Means, MEP.stdev)
# MEP.df$Names = factor(c("Pre 1", "Pre 2", "Pre 3", "Up 1", "Up 2", "Up 3", "Up 4",
#                          "Up 5", "Down 1", "Down 2", "Down 3", "Down 4", "Down 5"),
#                        levels = c("Pre 1", "Up 1", "Down 1", "Up 2", "Down 2",
#                                   "Pre 2", "Up 3", "Down 3", "Up 4", "Down 4",
#                                   "Pre 3", "Up 5", "Down 5"))
# MEP.df$Conditions = c('Pre', 'Pre', 'Pre', 'Up', 'Up', 'Up', 'Up', 'Up', 'Down',
#                       'Down', 'Down', 'Down', 'Down')
# MEP.df$Sess = c(1, 2, 3,
#                 1, 1, 2, 2, 3,
#                 1, 1, 2, 2, 3)
#
# MEP.df$Percentage.Change.MEP = MEP.df$MEP.Means / MEP.df$MEP.Means[MEP.df$Names == "Pre
1"]
# MEP.df$Percentage.Change.SEM = MEP.df$MEP.stdev / MEP.df$MEP.Means[MEP.df$Names == "Pre
1"]
#
# MEP.df$Percentage.Change.MEP.AllSess[MEP.df$Sess == 1] = MEP.df$MEP.Means[MEP.df$Sess ==
1] / MEP.df$MEP.Means[MEP.df$Names == "Pre 1"]
# MEP.df$Percentage.Change.MEP.AllSess[MEP.df$Sess == 2] = MEP.df$MEP.Means[MEP.df$Sess ==
2] / MEP.df$MEP.Means[MEP.df$Names == "Pre 2"]
# MEP.df$Percentage.Change.MEP.AllSess[MEP.df$Sess == 3] = MEP.df$MEP.Means[MEP.df$Sess ==
3] / MEP.df$MEP.Means[MEP.df$Names == "Pre 3"]
# MEP.df$Percentage.Change.SEM.AllSess[MEP.df$Sess == 1] = MEP.df$MEP.stdev[MEP.df$Sess ==
1] / MEP.df$MEP.Means[MEP.df$Names == "Pre 1"]
# MEP.df$Percentage.Change.SEM.AllSess[MEP.df$Sess == 2] = MEP.df$MEP.stdev[MEP.df$Sess ==
2] / MEP.df$MEP.Means[MEP.df$Names == "Pre 2"]
# MEP.df$Percentage.Change.SEM.AllSess[MEP.df$Sess == 3] = MEP.df$MEP.stdev[MEP.df$Sess ==
3] / MEP.df$MEP.Means[MEP.df$Names == "Pre 3"]
#
#
# Ind.part.boxplot = function(dataframe, participant, y.axis, x.axis){
#   dataframe = dataframe[dataframe$pid == participant, ]
#
#   ggplot(dataframe[dataframe$z.Transformation.Standardised.MEP < 3 &
dataframe$conditions %in% c("down", "up", "pre"), ],
#         aes(x = plot.Order,
#             y = actual_MEP,
#             color = conditions)) +
#     geom_point()
#
#   # facet_wrap(~ pid) +
#   # geom_flat_violin()

```

```

# # geom_jitter(width = .1) +
# # geom_smooth(method = lm)
# # geom_boxplot()
# #geom_violin(trim = TRUE)
#
#
# result = ggplot(data = dataframe, aes(x = x.axis,
#                                       y = y.axis,
#                                       fill = as.factor(conditions))) +
#   geom_boxplot(aes(fill = conditions))
# result
# }
#
# eLife.plot.rawVal = ggplot(MEP.df, aes(x = Names, y = MEP.Mean, color = Conditions)) +
#   geom_point() +
#   geom_errorbar(aes(ymin = MEP.Mean - MEP.stderr, ymax = MEP.Mean + MEP.stderr), width
# = .2,
#               position=position_dodge(0.05))
# eLife.plot.rawVal
#
# eLife.plot.PercChange = ggplot(MEP.df, aes(x = Names, y = Percentage.Change.MEP, color =
Conditions)) +
#   geom_point() +
#   geom_errorbar(aes(ymin = Percentage.Change.MEP - Percentage.Change.SEM, ymax =
Percentage.Change.MEP + Percentage.Change.SEM), width = .2,
#               position=position_dodge(0.05))
# eLife.plot.PercChange
#
# eLife.plot.PercChange.AllSess = ggplot(MEP.df, aes(x = Names, y =
Percentage.Change.MEP.AllSess, color = Conditions)) +
#   geom_point() +
#   geom_errorbar(aes(ymin = Percentage.Change.MEP.AllSess - Percentage.Change.SEM.AllSess,
ymax = Percentage.Change.MEP.AllSess + Percentage.Change.SEM.AllSess), width = .2,
#               position=position_dodge(0.05))
# eLife.plot.PercChange.AllSess

```

10.4.15 MEP Plots (R)

This script creates MEP related plots.

```
# This script creates MEP-related plots

# Initiation -----
#clear all
rm(list = ls())

library(ggplot2)
library(plotrix)

# Go into Folder
work_dir = ""
setwd(work_dir)
source(file = "")

load(file = "")

# exclude background rMS higher than 0.01
results = results[results$actual_rMS_right < 0.01, ]

# Plots -----
boxplot(results$actual_rMS_right ~ results$pid)
boxplot(results$z.Transformation.Standardised.MEP ~ interaction(results$pid,
results$conditions))

## Plot functions -----
dataframe = results

# plot order actually need to label pre from pre1 to pre3
create_MEP_framework = function(dataframe, MEPS.to.use){
  dataframe$plot.Order = interaction(dataframe$plot.blocks, dataframe$conditions)
  dataframe$plot.Order[dataframe$conditions == "pre"] =
interaction(dataframe$plot.sessions[dataframe$conditions == "pre"], "pre")
  dataframe$plot.Order = factor(dataframe$plot.Order, levels = c("1.pre", "1.down", "1.up",
"2.down", "2.up",
"2.pre", "3.down", "3.up",
"4.down", "4.up",
"3.pre", "5.down",
"5.up"))

  pre.mean.1 = mean(dataframe[[MEPS.to.use]][dataframe$plot.Order == "1.pre"], na.rm =
TRUE)
  pre.mean.2 = mean(dataframe[[MEPS.to.use]][dataframe$plot.Order == "2.pre"], na.rm =
TRUE)
  pre.mean.3 = mean(dataframe[[MEPS.to.use]][dataframe$plot.Order == "3.pre"], na.rm =
TRUE)
  up.mean.1 = mean(dataframe[[MEPS.to.use]][dataframe$plot.Order == "1.up"], na.rm = TRUE)
  up.mean.2 = mean(dataframe[[MEPS.to.use]][dataframe$plot.Order == "2.up"], na.rm = TRUE)
  up.mean.3 = mean(dataframe[[MEPS.to.use]][dataframe$plot.Order == "3.up"], na.rm = TRUE)
  up.mean.4 = mean(dataframe[[MEPS.to.use]][dataframe$plot.Order == "4.up"], na.rm = TRUE)
  up.mean.5 = mean(dataframe[[MEPS.to.use]][dataframe$plot.Order == "5.up"], na.rm = TRUE)
  down.mean.1 = mean(dataframe[[MEPS.to.use]][dataframe$plot.Order == "1.down"], na.rm =
TRUE)
  down.mean.2 = mean(dataframe[[MEPS.to.use]][dataframe$plot.Order == "2.down"], na.rm =
TRUE)
  down.mean.3 = mean(dataframe[[MEPS.to.use]][dataframe$plot.Order == "3.down"], na.rm =
TRUE)
  down.mean.4 = mean(dataframe[[MEPS.to.use]][dataframe$plot.Order == "4.down"], na.rm =
TRUE)
  down.mean.5 = mean(dataframe[[MEPS.to.use]][dataframe$plot.Order == "5.down"], na.rm =
TRUE)
  MEP.Mean = c(pre.mean.1, pre.mean.2, pre.mean.3, up.mean.1, up.mean.2, up.mean.3,
```

```
up.mean.4, up.mean.5, down.mean.1, down.mean.2, down.mean.3, down.mean.4,
down.mean.5)
```

```
pre.stderr.1 = std.error(dataframe[[MEPs.to.use]][dataframe$plot.Order == "1.pre"], na.rm
= TRUE)
pre.stderr.2 = std.error(dataframe[[MEPs.to.use]][dataframe$plot.Order == "2.pre"], na.rm
= TRUE)
pre.stderr.3 = std.error(dataframe[[MEPs.to.use]][dataframe$plot.Order == "3.pre"], na.rm
= TRUE)
up.stderr.1 = std.error(dataframe[[MEPs.to.use]][dataframe$plot.Order == "1.up"], na.rm =
TRUE)
up.stderr.2 = std.error(dataframe[[MEPs.to.use]][dataframe$plot.Order == "2.up"], na.rm =
TRUE)
up.stderr.3 = std.error(dataframe[[MEPs.to.use]][dataframe$plot.Order == "3.up"], na.rm =
TRUE)
up.stderr.4 = std.error(dataframe[[MEPs.to.use]][dataframe$plot.Order == "4.up"], na.rm =
TRUE)
up.stderr.5 = std.error(dataframe[[MEPs.to.use]][dataframe$plot.Order == "5.up"], na.rm =
TRUE)
down.stderr.1 = std.error(dataframe[[MEPs.to.use]][dataframe$plot.Order == "1.down"],
na.rm = TRUE)
down.stderr.2 = std.error(dataframe[[MEPs.to.use]][dataframe$plot.Order == "2.down"],
na.rm = TRUE)
down.stderr.3 = std.error(dataframe[[MEPs.to.use]][dataframe$plot.Order == "3.down"],
na.rm = TRUE)
down.stderr.4 = std.error(dataframe[[MEPs.to.use]][dataframe$plot.Order == "4.down"],
na.rm = TRUE)
down.stderr.5 = std.error(dataframe[[MEPs.to.use]][dataframe$plot.Order == "5.down"],
na.rm = TRUE)
MEP.stderr = c(pre.stderr.1, pre.stderr.2, pre.stderr.3, up.stderr.1, up.stderr.2,
up.stderr.3, up.stderr.4, up.stderr.5, down.stderr.1, down.stderr.2,
down.stderr.3, down.stderr.4, down.stderr.5)
MEP.df = data.frame(MEP.Means, MEP.stderr)
MEP.df$Names = factor(c("Pre 1", "Pre 2", "Pre 3", "Up 1", "Up 2", "Up 3", "Up 4",
"Up 5", "Down 1", "Down 2", "Down 3", "Down 4", "Down 5"),
levels = c("Pre 1", "Up 1", "Down 1", "Up 2", "Down 2",
"Pre 2", "Up 3", "Down 3", "Up 4", "Down 4",
"Pre 3", "Up 5", "Down 5"))
MEP.df$Conditions = c("Pre", "Pre", "Pre", "Up", "Up", "Up", "Up", "Up", "Up", "Down",
"Down", "Down", "Down", "Down")
MEP.df$Sess = c(1, 2, 3,
1, 1, 2, 2, 3,
1, 1, 2, 2, 3)

MEP.df$Percentage.Change.MEP = MEP.df$MEP.Means / MEP.df$MEP.Means[MEP.df$Names == "Pre
1"]
MEP.df$Percentage.Change.SEM = MEP.df$MEP.stderr / MEP.df$MEP.Means[MEP.df$Names == "Pre
1"]

MEP.df$Percentage.Change.MEP.AllSess[MEP.df$Sess == 1] = MEP.df$MEP.Means[MEP.df$Sess ==
1] / MEP.df$MEP.Means[MEP.df$Names == "Pre 1"]
MEP.df$Percentage.Change.MEP.AllSess[MEP.df$Sess == 2] = MEP.df$MEP.Means[MEP.df$Sess ==
2] / MEP.df$MEP.Means[MEP.df$Names == "Pre 2"]
MEP.df$Percentage.Change.MEP.AllSess[MEP.df$Sess == 3] = MEP.df$MEP.Means[MEP.df$Sess ==
3] / MEP.df$MEP.Means[MEP.df$Names == "Pre 3"]
MEP.df$Percentage.Change.SEM.AllSess[MEP.df$Sess == 1] = MEP.df$MEP.stderr[MEP.df$Sess ==
1] / MEP.df$MEP.Means[MEP.df$Names == "Pre 1"]
MEP.df$Percentage.Change.SEM.AllSess[MEP.df$Sess == 2] = MEP.df$MEP.stderr[MEP.df$Sess ==
2] / MEP.df$MEP.Means[MEP.df$Names == "Pre 2"]
MEP.df$Percentage.Change.SEM.AllSess[MEP.df$Sess == 3] = MEP.df$MEP.stderr[MEP.df$Sess ==
3] / MEP.df$MEP.Means[MEP.df$Names == "Pre 3"]
MEP.df$PlotBlock = c("BS", "BS", "BS",
"B1", "B2", "B1", "B2", "B1",
"B1", "B2", "B1", "B2", "B1")
MEP.df$`Session and Condition` = interaction(MEP.df$PlotBlock, MEP.df$Sess, sep = " S")
MEP.df$`Session and Condition` = factor(MEP.df$`Session and Condition`, levels = c("BS
S1", "B1 S1", "B2 S1",
```



```

S2', 'B1 S2', 'B2 S2',
S3', 'B1 S3'))
MEP.df$upline = NA
MEP.df$upline[(MEP.df$Sess == 1 & (MEP.df$Conditions %in% c("Up", "Pre")))] = "Up1"
MEP.df$upline[(MEP.df$Sess == 2 & (MEP.df$Conditions %in% c("Up", "Pre")))] = "Up2"
MEP.df$upline[(MEP.df$Sess == 3 & (MEP.df$Conditions %in% c("Up", "Pre")))] = "Up3"
MEP.df$downline = NA
MEP.df$downline[(MEP.df$Sess == 1 & (MEP.df$Conditions %in% c("Down", "Pre")))] = "Down1"
MEP.df$downline[(MEP.df$Sess == 2 & (MEP.df$Conditions %in% c("Down", "Pre")))] = "Down2"
MEP.df$downline[(MEP.df$Sess == 3 & (MEP.df$Conditions %in% c("Down", "Pre")))] = "Down3"

return(MEP.df)
}

# legacy plot functions
{
  elife.plot.rawMEPS = function(MEP.summary.df){
    gg.plot = ggplot(MEP.summary.df, aes(x = Names, y = MEP.Mean, color = Conditions)) +
      geom_point() +
      geom_errorbar(aes(ymin = MEP.Mean - MEP.stderr, ymax = MEP.Mean + MEP.stderr),
width = .2,
position=position_dodge(0.05))

    return(gg.plot)
  }

  elife.plot.percChange = function(MEP.summary.df){
    gg.plot = ggplot(MEP.df, aes(x = Names, y = Percentage.Change.MEP, color = Conditions))
+
      geom_point() +
      geom_errorbar(aes(ymin = Percentage.Change.MEP - Percentage.Change.SEM, ymax =
Percentage.Change.MEP + Percentage.Change.SEM), width = .2,
position=position_dodge(0.05))

    return(gg.plot)
  }

  elife.plot.percChange.AllSess = function(MEP.summary.df){
    gg.plot = elife.plot.PercChange.AllSess = ggplot(MEP.df, aes(x = Names, y =
Percentage.Change.MEP.AllSess, color = Conditions)) +
      geom_point() +
      theme_light() +
      geom_errorbar(aes(ymin = Percentage.Change.MEP.AllSess -
Percentage.Change.SEM.AllSess, ymax = Percentage.Change.MEP.AllSess +
Percentage.Change.SEM.AllSess), width = .2,
position=position_dodge(0.05))
    elife.plot.PercChange.AllSess

    return(gg.plot)
  }
}

df.MEP.raw = create_MEP_framework(dataframe, 'actual_MEP')
df.MEP.dyn = create_MEP_framework(dataframe, 'Dynamic.Standardised.MEP')

Ind.part.boxplot = function(dataframe, participant, y.axis, x.axis){
  dataframe = dataframe[dataframe$pid == participant, ]

  ggplot(dataframe[dataframe$z.Transformation.Standardised.MEP < 3 &
dataframe$conditions %in% c("down", "up", "pre"), ],
aes(x = plot.Order,
y = actual_MEP,
color = conditions)) +
  geom_point()

# facet_wrap(~ pid) +

```

```

# geom_flat_violin()
# geom_jitter(width = .1) +
# geom_smooth(method = lm)
# geom_boxplot()
#geom_violin(trim = TRUE)

result = ggplot(data = dataframe, aes(x = x.axis,
                                      y = y.axis,
                                      fill = as.factor(conditions))) +
  geom_boxplot(aes(fill = conditions))
result
}

# Plot Plots -----

elif.plot.sess.MEPMeans = function(mep.df){
  gg.plot = ggplot(mep.df, aes(x = `Session and Condition`, y = MEP.Means, color =
Conditions, fill = Conditions)) +
    geom_point(aes(shape = Conditions), size = 8) +
    geom_errorbar(aes(ymin = MEP.Means - MEP.stderr, ymax = MEP.Means + MEP.stderr), width
= .2) +
    theme_light() +
    labs(y = "MEP Means") +
    scale_shape_manual(values=c(25, 19, 24)) +
    scale_colour_viridis_d(alpha = 0.8) +
    scale_fill_viridis_d(alpha = 0.8)
  return(gg.plot)
}

elif.plot.sess.percChange = function(mep.df){
  gg.plot = ggplot(mep.df, aes(x = `Session and Condition`, y = Percentage.Change.MEP,
color = Conditions, fill = Conditions)) +
    geom_point(aes(shape = Conditions), size = 8) +
    geom_errorbar(aes(ymin = Percentage.Change.MEP - Percentage.Change.SEM, ymax =
Percentage.Change.MEP + Percentage.Change.SEM), width = .2) +
    theme_light() +
    labs(y = "MEP Percentage change from first Baseline") +
    scale_shape_manual(values=c(25, 19, 24)) +
    scale_colour_viridis_d(alpha = 0.8) +
    scale_fill_viridis_d(alpha = 0.8)
  return(gg.plot)
}

elif.plot.sess.percChange.Allsess = function(mep.df){
  mep.df$Conditions = factor(mep.df$Conditions, levels = c("Up", "Pre", "Down"))
  levels(mep.df$Conditions)[levels(mep.df$Conditions)=='Pre'] <- 'Baseline'
  gg.plot = ggplot(mep.df, aes(x = `Session and Condition`, y =
Percentage.Change.MEP.AllSess, color = Conditions, fill = Conditions)) +
    geom_errorbar(aes(ymin = Percentage.Change.MEP.AllSess - Percentage.Change.SEM.AllSess,
ymax = Percentage.Change.MEP.AllSess + Percentage.Change.SEM.AllSess), width = .2) +
    theme_light() +
    geom_line(data = subset(mep.df, downline == "Down1"), aes(group = downline), color =
"#440154", alpha = 0.8) +
    geom_line(data = subset(mep.df, downline == "Down2"), aes(group = downline), color =
"#440154", alpha = 0.8) +
    geom_line(data = subset(mep.df, downline == "Down3"), aes(group = downline), color =
"#440154", alpha = 0.8) +
    geom_line(data = subset(mep.df, upline == "Up1"), aes(group = upline), color =
"#fde725", alpha = 0.8) +
    geom_line(data = subset(mep.df, upline == "Up2"), aes(group = upline), color =
"#fde725", alpha = 0.8) +
    geom_line(data = subset(mep.df, upline == "Up3"), aes(group = upline), color =
"#fde725", alpha = 0.8) +
    geom_point(aes(shape = Conditions), size = 5) +
    labs(y = "MEP Percentage Change from Their Respective Baseline") +
    theme(text = element_text(family = "Times"),

```

```

axis.text.x = element_text(angle = 25, hjust = 1)) +
  scale_x_discrete(labels=c("BS S1" = "Baseline Session 1", "B1 S1" = "Session 1 Block
1",
                                "B2 S1" = "Session 1 Block 2", "BS S2" = "Baseline Session
2",
                                "B1 S2" = "Session 2 Block 1", "B2 S2" = "Session 2 Block 2",
                                "BS S3" = "Baseline EEG Session", "B1 S3" = "EEG Session
Block 1")) +
  scale_shape_manual(values=c(24, 19, 25)) +
  scale_colour_viridis_d(alpha = 0.8, direction = -1) +
  scale_fill_viridis_d(alpha = 0.8, direction = -1)
return(gg.plot)
}

elife.plot.raw = function(mep.df){
  mep.df$Conditions = factor(mep.df$Conditions, levels = c("Up", "Pre", "Down"))
  levels(mep.df$Conditions)[levels(mep.df$Conditions)=='Pre'] <- 'Baseline'
  gg.plot = ggplot(mep.df, aes(x = `Session and Condition`, y = MEP.Mean, color =
Conditions, fill = Conditions)) +
    geom_errorbar(aes(ymin = MEP.Mean - MEP.stderr, ymax = MEP.Mean + MEP.stderr), width
= .2) +
    theme_light() +
    geom_line(data = subset(mep.df, downline == "Down1"), aes(group = downline), color =
"#440154", alpha = 0.8) +
    geom_line(data = subset(mep.df, downline == "Down2"), aes(group = downline), color =
"#440154", alpha = 0.8) +
    geom_line(data = subset(mep.df, downline == "Down3"), aes(group = downline), color =
"#440154", alpha = 0.8) +
    geom_line(data = subset(mep.df, upline == "Up1"), aes(group = upline), color =
"#fde725", alpha = 0.8) +
    geom_line(data = subset(mep.df, upline == "Up2"), aes(group = upline), color =
"#fde725", alpha = 0.8) +
    geom_line(data = subset(mep.df, upline == "Up3"), aes(group = upline), color =
"#fde725", alpha = 0.8) +
    geom_point(aes(shape = Conditions), size = 5) +
    labs(y = "MEP Percentage Change from Their Respective Baseline") +
    theme(text = element_text(family = "Times"),
          axis.text.x = element_text(angle = 25, hjust = 1)) +
    scale_x_discrete(labels=c("BS S1" = "Baseline Session 1", "B1 S1" = "Session 1 Block
1",
                                "B2 S1" = "Session 1 Block 2", "BS S2" = "Baseline Session
2",
                                "B1 S2" = "Session 2 Block 1", "B2 S2" = "Session 2 Block 2",
                                "BS S3" = "Baseline EEG Session", "B1 S3" = "EEG Session
Block 1")) +
    scale_shape_manual(values=c(24, 19, 25)) +
    scale_colour_viridis_d(alpha = 0.8, direction = -1) +
    scale_fill_viridis_d(alpha = 0.8, direction = -1)
  return(gg.plot)
}

# do some stats: (technically all of that is in LinMods)
# first session block one difference up down
S1 = list()
S2 = list()
S3 = list()
S1$B1 = MEP_Test_Block(dataframe, sess = "1", block = "1",
                      testname = "Ptest sess1 block1 UP v DOWN",
                      directionality = "greater")
# Save Data and Save Results
MEP_Test_Block_Save(Data = S1$B1, SaveDir = "../Processed Data/MEP Data/Stats/", SaveName =
"Sess1-Block1-UpVDown")
# Block2
S1$B2 = MEP_Test_Block(dataframe, sess = "1", block = "2",
                      testname = "Ptest sess1 block2 UP v DOWN",
                      directionality = "greater")
# Save Data and Save Results

```

```

MEP_Test_Block_Save(Data = S1$B2, SaveDir = "./Processed Data/MEP Data/Stats/", SaveName =
"Sess1-Block2-UpVDown")

S2$B3 = MEP_Test_Block(dataframe, sess = "2", block = "3",
                        testname = "Ptest sess2 block1 UP v DOWN",
                        directionality = "greater")
# Save Data and Save Results
MEP_Test_Block_Save(Data = S2$B3, SaveDir = "./Processed Data/MEP Data/Stats/", SaveName =
"Sess2-Block1-UpVDown")

S2$B4 = MEP_Test_Block(dataframe, sess = "2", block = "4",
                        testname = "Ptest sess2 block2 UP v DOWN",
                        directionality = "greater")
# Save Data and Save Results
MEP_Test_Block_Save(Data = S2$B4, SaveDir = "./Processed Data/MEP Data/Stats/", SaveName =
"Sess2-Block2-UpVDown")

S3$B5 = MEP_Test_Block(dataframe, sess = "3", block = "5",
                        testname = "Ptest sess3 block1 UP v DOWN",
                        directionality = "greater")
# Save Data and Save Results
MEP_Test_Block_Save(Data = S3$B5, SaveDir = "./Processed Data/MEP Data/Stats/", SaveName =
"Sess3-Block1-UpVDown")

# Main Plot
raw.elife.plot.ssess.percchange.allsess = elife.plot.ssess.percChange.Allsess(df.MEP.raw)
raw.elife.plot.ssess.percchange.allsess
ggsave(filename = './Processed Data/MEP
Data/H1/Plots/raw.elife.plot.ssess.percchange.alsess.pdf', width = 8, height = 4.5)

# Main plot but with raw values
raw.plot.mepsem = elife.plot.raw(df.MEP.raw)
raw.plot.mepsem
ggsave(filename = './Processed Data/MEP Data/H1/Plots/raw.plot.mepsem.pdf', width = 8,
height = 4.5)

# # Main Plot that one does not make much sense, as dynamic is already a kind of percentage
transformation
# dyn.elife.plot.ssess.percchange.allsess = elife.plot.ssess.percChange.Allsess(df.MEP.dyn)
# dyn.elife.plot.ssess.percchange.allsess
# ggsave(filename = './Processed Data/MEP
Data/H1/Plots/dyn.elife.plot.ssess.percchange.alsess.pdf', width = 8, height = 4.5)

# Main plot but with dyn values
dyn.plot.mepsem = elife.plot.raw(df.MEP.dyn)
dyn.plot.mepsem
ggsave(filename = './Processed Data/MEP Data/H1/Plots/dyn.plot.mepsem.pdf', width = 8,
height = 4.5)

# #out of interest, I will do same with my dynamic measure
# dyn.elife.plot.ssess.percchange.allsess = elife.plot.ssess.percChange.Allsess(df.MEP.dyn)
# dyn.elife.plot.ssess.percchange.allsess
# ggsave(filename = './Processed Data/MEP
Data/H1/Plots/dyn.elife.plot.ssess.percchange.allsess.pdf', width = 8, height = 4.5)
#
# dyn.plot.mepsem = elife.plot.raw(df.MEP.dyn)
# dyn.plot.mepsem
# ggsave(filename = './Processed Data/MEP Data/H1/Plots/dyn.plot.mepsem.pdf', width = 8,
height = 4.5)

```

10.4.16 Stability and Distance (R)

This script does of a lot of checks with the stability and distinctiveness metrics.

```
# This script we will look at the StabDist Metrics

# Init -----
# clear all
rm(list = ls())

# Libraries
library(lmerTest)
library(lme4)
library(doParallel)
library(ggplot2)
library(extrafont)
library(ggpubr)
library(see)
library(readxl)

# set wd
work_dir = ""
setwd(work_dir)

# get useful functions
source(file = "")

df.StabDist = read_excel("./Processed
Data/StabDist/Processed/Stability_Distance_Results.xlsx")

# Data Adjustements -----

# Continuous block numbers
df.StabDist$Continuous.Block = (df.StabDist$Session - 1) * 4 + df.StabDist$Block

# Delete any PID = 0
df.StabDist = df.StabDist[!df.StabDist$Continuous.Block == -4, ]
check_summary(df.StabDist$Continuous.Block)

# Add Groups
TMS.PIDs = c('')
df.StabDist$Group = "EEG"
df.StabDist$Group[df.StabDist$PID %in% TMS.PIDs] = "TMS"
remove(TMS.PIDs)
table(df.StabDist$Group)

# Block with only 9-20
df.eegbci = df.StabDist[df.StabDist$Session %in% c(3, 4, 5), ]

# Quality Check -----

# summarise trials
check_summary(df.StabDist$`# of Up`)
check_summary(df.StabDist$`# of Trials`)
check_summary(df.StabDist$`# of Down`)
trial.nr = dim(df.StabDist)[1]

## Investigate Trials -----
# Investigate Trials
balancing = df.StabDist$`# of Up` / df.StabDist$`# of Down`
balancing.sd = 3*sd(balancing)
balancing.mean = mean(balancing)

# first let's check if they are different
upvsdown.ttest = t.test(df.StabDist$`# of Up`, (df.StabDist$`# of Down`), alternative =
"two.sided", paired = TRUE)
```

```

# better test but same outcome so just use it for documentation
upvsdown.ttest.doc = PairedTest(df.StabDist$`# of Up`, (df.StabDist$`# of Down`), testName
= "Up Trials vs Down Trials for Stability & Distinctiveness analysis")
write(capture.output(upvsdown.ttest.doc), file = paste("../Processed
Data/StabDist/Quality/Trials/Upvsdowntest.txt"))

upvsdown.pval = upvsdown.ttest$p.value
# they are not, is it significantly different from 0.5 though?
upvsdown.m <- lm(balancing ~ 1, data = df.StabDist)
# Calculate the confidence interval
confint(upvsdown.m, level=0.95)

pdf(file = "../Processed Data/StabDist/Quality/Trials/UpVDn-plot.pdf")
{
  plot(balancing, x = df.StabDist$PID,
       ylim = c(balancing.mean-2*balancing.sd,
                 balancing.mean+2*balancing.sd),
       ylab = "(up / down) trials",
       xlab = "PID")
  abline(h = balancing.mean+balancing.sd, col = "blue")
  abline(h = balancing.mean, col = "red")
  abline(h = balancing.mean-balancing.sd, col = "blue")
}
dev.off()

remove(upvsdown.m, upvsdown.pval, upvsdown.ttest, upvsdown.ttest.doc,
       balancing, balancing.mean, balancing.sd)

# Stability Outliers -----
# Rest
Stability.Outliers = Check_For_Outliers(chunk.data.y = df.StabDist$`Stability of Rest`,
                                       pdf.path.name = "../Processed
Data/StabDist/Quality/Stability/Rest-outliers.pdf")
df.StabDist$Stab.Rest.Outliers = Stability.Outliers$outliers
# Up
Stability.Outliers = Check_For_Outliers(chunk.data.y = df.StabDist$`Stability of UP`,
                                       pdf.path.name = "../Processed
Data/StabDist/Quality/Stability/Up-outliers.pdf")
df.StabDist$Stab.Up.Outliers = Stability.Outliers$outliers
# Down
Stability.Outliers = Check_For_Outliers(chunk.data.y = df.StabDist$`Stability of DN`,
                                       pdf.path.name = "../Processed
Data/StabDist/Quality/Stability/Down-outliers.pdf")
df.StabDist$Stab.Down.Outliers = Stability.Outliers$outliers

df.StabDist$Stab.Outliers = df.StabDist$Stab.Up.Outliers | df.StabDist$Stab.Down.Outliers |
df.StabDist$Stab.Rest.Outliers

# Distance Outliers -----
# Rest
Distance.Outliers = Check_For_Outliers(chunk.data.y = df.StabDist$`Distance UP:DN`,
                                       pdf.path.name = "../Processed
Data/StabDist/Quality/Distance/Dist-UpDn-outliers.pdf")
df.StabDist$Dist.UpDn.Outliers = Distance.Outliers$outliers
# Up
Distance.Outliers = Check_For_Outliers(chunk.data.y = df.StabDist$`Distance Rest:UP`,
                                       pdf.path.name = "../Processed
Data/StabDist/Quality/Distance/Dist-RestUp-outliers.pdf")
df.StabDist$Dist.RestUp.Outliers = Distance.Outliers$outliers
# Down
Distance.Outliers = Check_For_Outliers(chunk.data.y = df.StabDist$`Distance Rest:DN`,
                                       pdf.path.name = "../Processed
Data/StabDist/Quality/Distance/Dist-RestDn-outliers.pdf")
df.StabDist$Dist.RestDown.Outliers = Distance.Outliers$outliers

df.StabDist$Dist.Outliers = df.StabDist$Dist.RestUp.Outliers |
df.StabDist$Dist.RestDown.Outliers | df.StabDist$Dist.UpDn.Outliers

```

```

# Incorporate Achievers -----
load(file = "./Processed Data/CA Data/df-Achievers.R")
df.StabDist$UID = interaction(df.StabDist$PID, df.StabDist$Session, df.StabDist$Block, sep
= ".")

# nr 10 not in here at all and nr 67 "falls out", fixed both
df.StabDist.complete = merge(df.StabDist, df.achievers,
                             by = intersect(names(df.StabDist), names(df.achievers)))

save(df.StabDist.complete, file = "./Processed Data/StabDist/Processed/CA-StabDist-
Achievers.R")

```

10.4.17 Stability and Distance 2 (R)

This script does all the testing on the distance and stability metrics.

```
# Initialisation -----

# clear all
rm(list = ls())

# Libraries
library(ggplot2)
library(ggh4x)
library(see)
library(lme4)
library(lmerTest)
library(car)
library(lmtest)

# set wd
work_dir = "~/Library/CloudStorage/OneDrive-TrinityCollegeDublin/2 Stage BCI Sync
Folder/Complete Data/230221 Analysis/"
setwd(work_dir)

# get useful functions
source(file = "../..../Code/R Code/R_useful_funcs_share.R")

# Read in data:
load("../Processed Data/StabDist/Processed/CA-StabDist-Achievers.R")
df.StabDist = df.StabDist.complete
# take out number 9
df.StabDist = df.StabDist[df.StabDist$PID != 9, ]

# Some NA CA?
df.StabDist = df.StabDist[!is.na(df.StabDist$CA), ]
# # blocks are the same so I'll take one and discard the rest
# df.StabDist[["Block.y"]] = NULL
# colnames(df.StabDist)[colnames(df.StabDist) == "Block.x"] = "Block"
# Take out some details
df.StabDist = df.StabDist[!df.StabDist$Block == 0, ]
df.EEGBCI = df.StabDist[df.StabDist$Sess %in% c(3, 4, 5), ]
df.EEGBCI.KSS = df.EEGBCI[!is.na(df.EEGBCI$KSS), ]

trial.nr = dim(df.StabDist)[1]

# First Look at Models -----
formula.base = formula(CA ~ Continuous.Block * Group + (1|PID))
save_dir = "../Processed Data/StabDist/Models/Base"
lm.base = Basic_LM(lm.data = df.EEGBCI, formula.base, save_path = save_dir)
classic.CA.Plot(df.EEGBCI, lm.model = lm.base$Model, save_path = save_dir)

# Stability of Rest -----
StabRest.ri = Model_me_this(df.EEGBCI,
                           "`Stability of Rest` ~ Continuous.Block * Group+ (1|PID)",
                           "../Processed Data/StabDist/Models/Stability/Rest/Random
Intercept/StabDist-Ri")

StabRest.ri = Model_me_this(df.EEGBCI.KSS,
                           "`Stability of Rest` ~ Continuous.Block * Group + KSS +
(1|PID)",
                           "../Processed Data/StabDist/Models/Stability/Rest/KSS Random
Intercept/StabDist-Ri")

StabRest.risl = Model_me_this(df.EEGBCI,
                              "`Stability of Rest` ~ Continuous.Block * Group+
(Continuous.Block|PID)",
```



```

                                "./Processed Data/StabDist/Models/Stability/Rest/Random
Slope/StabDist-RiSl")

StabRest.risl = Model_me_this(df.EEGBCI.KSS,
                                "`Stability of Rest` ~ Continuous.Block * Group + KSS +
(Continuous.Block|PID)",
                                "./Processed Data/StabDist/Models/Stability/Rest/KSS Random
Slope/StabDist-RiSl")

StabRest.Comparison = anova(StabRest.ri$LM$Model, StabRest.risl$LM$Model)
write(capture.output(StabRest.Comparison), file = "./Processed
Data/StabDist/Models/Stability/Rest/Rest-ri-risl-comparison.txt")

# Stability of Up -----
StabUp.ri = Model_me_this(df.EEGBCI,
                                "`Stability of UP` ~ Continuous.Block * Group+ (1|PID)",
                                "./Processed Data/StabDist/Models/Stability/Up/Random
Intercept/StabDist-Ri")

StabUp.ri = Model_me_this(df.EEGBCI.KSS,
                                "`Stability of UP` ~ Continuous.Block * Group + KSS + (1|PID)",
                                "./Processed Data/StabDist/Models/Stability/Up/KSS Random
Intercept/StabDist-Ri")

StabUp.risl = Model_me_this(df.EEGBCI,
                                "`Stability of UP` ~ Continuous.Block * Group +
(Continuous.Block|PID)",
                                "./Processed Data/StabDist/Models/Stability/Up/Random
Slope/StabDist-RiSl")

StabUp.risl = Model_me_this(df.EEGBCI.KSS,
                                "`Stability of UP` ~ Continuous.Block * Group + KSS +
(Continuous.Block|PID)",
                                "./Processed Data/StabDist/Models/Stability/Up/KSS Random
Slope/StabDist-RiSl")

StabUp.Comparison = anova(StabUp.ri$LM$Model, StabUp.risl$LM$Model)
write(capture.output(StabUp.Comparison), file = "./Processed
Data/StabDist/Models/Stability/Up/Up-ri-risl-comparison.txt")

# Stability of Down -----
StabDown.ri = Model_me_this(df.EEGBCI,
                                "`Stability of DN` ~ Continuous.Block * Group+ (1|PID)",
                                "./Processed Data/StabDist/Models/Stability/Down/Random
Intercept/StabDist-Ri")

StabDown.ri = Model_me_this(df.EEGBCI.KSS,
                                "`Stability of DN` ~ Continuous.Block * Group + KSS + (1|PID)",
                                "./Processed Data/StabDist/Models/Stability/Down/KSS Random
Intercept/StabDist-Ri")

StabDown.risl = Model_me_this(df.EEGBCI,
                                "`Stability of DN` ~ Continuous.Block * Group +
(Continuous.Block|PID)",
                                "./Processed Data/StabDist/Models/Stability/Down/Random
Slope/StabDist-RiSl")

StabDown.risl = Model_me_this(df.EEGBCI.KSS,
                                "`Stability of DN` ~ Continuous.Block * Group + KSS +
(Continuous.Block|PID)",
                                "./Processed Data/StabDist/Models/Stability/Down/KSS Random
Slope/StabDist-RiSl")

StabDown.Comparison = anova(StabDown.ri$LM$Model, StabDown.risl$LM$Model)
write(capture.output(StabDown.Comparison), file = "./Processed
Data/StabDist/Models/Stability/Down/Down-ri-risl-comparison.txt")

# Stability and CA -----

```

```

formula.baseCA = formula(CA ~ Continuous.Block * Group + (1|PID))
save_dir = "./Processed Data/StabDist/Models/Stability/CA-base"
lm.base.ca = Basic_LM(lm.data = df.EEGBCI, formula.baseCA, save_path = save_dir)
classic.CA.Plot(df.EEGBCI, lm.model = lm.base$Model, save_path = save_dir)

formula.StabDNCA = formula(CA ~ Continuous.Block * Group + `Stability of DN` + `Stability
of DN`:Group + (1|PID))
save_dir = "./Processed Data/StabDist/Models/Stability/CA-StabDN"
lm.base = Basic_LM(lm.data = df.EEGBCI, formula.StabDNCA, save_path = save_dir)
classic.CA.Plot(df.EEGBCI, lm.model = lm.base$Model, save_path = save_dir)

formula.StabUPCA = formula(CA ~ Continuous.Block * Group + `Stability of UP` + `Stability
of UP`:Group + (1|PID))
save_dir = "./Processed Data/StabDist/Models/Stability/CA-StabUP"
lm.base.up = Basic_LM(lm.data = df.EEGBCI, formula.StabUPCA, save_path = save_dir)
classic.CA.Plot(df.EEGBCI, lm.model = lm.base$Model, save_path = save_dir)

formula.StabUPCA.comp = formula(CA ~ Continuous.Block * Group * `Stability of UP` +
(1|PID))
save_dir = "./Processed Data/StabDist/Models/Stability/CA-StabUP-Complete"
lm.base.upcomp = Basic_LM(lm.data = df.EEGBCI, formula.StabUPCA.comp, save_path = save_dir)
classic.CA.Plot(df.EEGBCI, lm.model = lm.base$Model, save_path = save_dir)

formula.StabRestCA = formula(CA ~ Continuous.Block * Group + `Stability of Rest` +
`Stability of Rest`:Group + (1|PID))
save_dir = "./Processed Data/StabDist/Models/Stability/CA-StabRest"
lm.base = Basic_LM(lm.data = df.EEGBCI, formula.StabRestCA, save_path = save_dir)
classic.CA.Plot(df.EEGBCI, lm.model = lm.base$Model, save_path = save_dir)

ComparisonCA = anova(lm.base.upcomp$Model, lm.base.up$Model, lm.base.ca$Model)
write(capture.output(ComparisonCA), file = "./Processed Data/StabDist/Models/Stability/UP-
Comparison.txt")

# Stability and CA and KSS -----
Model_me_this(df.EEGBCI.KSS,
              "CA ~ Continuous.Block * Group + `Stability of UP` * KSS + (1|PID)",
              "./Processed Data/StabDist/Models/Stability/UP/CA-KSS/CA-KSS-Up")

Model_me_this(df.EEGBCI.KSS,
              "CA ~ Continuous.Block * Group + `Stability of DN` * KSS + (1|PID)",
              "./Processed Data/StabDist/Models/Stability/Down/CA-KSS/CA-KSS-DN")

Model_me_this(df.EEGBCI.KSS,
              "CA ~ Continuous.Block * Group + `Stability of Rest` * KSS + (1|PID)",
              "./Processed Data/StabDist/Models/Stability/Rest/CA-KSS/CA-KSS-Rest")

Model_me_this(df.EEGBCI.KSS,
              "CA ~ Continuous.Block * Group + `Stability of Rest` * KSS +
Continuous.Block:`Stability of Rest` + (1|PID)",
              "./Processed Data/StabDist/Models/Stability/Rest/CA-KSS-Cont.Block/CA-KSS-
ContBlock-Rest")

# Distance of UPvDN -----
DistUpvDn.ri = Model_me_this(df.EEGBCI,
                             "`Distance UP:DN` ~ Continuous.Block * Group + (1|PID)",
                             "./Processed Data/StabDist/Models/Distinctiveness/UPvDN/Random
Intercept/DistUpvDn-Ri")

DistUpvDn.ri.kss = Model_me_this(df.EEGBCI.KSS,
                                 "`Distance UP:DN` ~ Continuous.Block * Group + KSS +
(1|PID)",
                                 "./Processed
Data/StabDist/Models/Distinctiveness/UPvDN/KSS Random Intercept/DistUpvDn-Ri")

DistUpvDn.risl = Model_me_this(df.EEGBCI,
                                "`Distance UP:DN` ~ Continuous.Block * Group +
(Continuous.Block|PID)",
                                )

```

```

                                "./Processed
Data/StabDist/Models/Distinctiveness/UPvDN/Random Slope/StabDist-RiSl")

DistUpvDn.risl.kss = Model_me_this(df.EEGBCI.KSS,
                                "`Distance UP:DN` ~ Continuous.Block * Group + KSS +
(Continuous.Block|PID)",
                                "./Processed
Data/StabDist/Models/Distinctiveness/UPvDN/KSS Random Slope/StabDist-RiSl")

DistUPvDn.Comparison = anova(DistUpvDn.ri$LM$Model, DistUpvDn.risl$LM$Model)
write(capture.output(DistUPvDn.Comparison), file = "./Processed
Data/StabDist/Models/Distinctiveness/UPvDN/UPvDN-ri-risl-comparison.txt")

# Distance of UPvRest -----
DistUPvRest.ri = Model_me_this(df.EEGBCI,
                                "`Distance Rest:UP` ~ Continuous.Block * Group + (1|PID)",
                                "./Processed
Data/StabDist/Models/Distinctiveness/UPvRest/Random Intercept/DistUPvRest-Ri")

DistUPvRest.ri.kss = Model_me_this(df.EEGBCI.KSS,
                                "`Distance Rest:UP` ~ Continuous.Block * Group + KSS +
(1|PID)",
                                "./Processed
Data/StabDist/Models/Distinctiveness/UPvRest/KSS Random Intercept/DistUPvRest-Ri")

DistUPvRest.risl = Model_me_this(df.EEGBCI,
                                "`Distance Rest:UP` ~ Continuous.Block * Group +
(Continuous.Block|PID)",
                                "./Processed
Data/StabDist/Models/Distinctiveness/UPvRest/Random Slope/DistUPvRest-RiSl")

DistUPvRest.risl.kss = Model_me_this(df.EEGBCI.KSS,
                                "`Distance Rest:UP` ~ Continuous.Block * Group + KSS +
(Continuous.Block|PID)",
                                "./Processed
Data/StabDist/Models/Distinctiveness/UPvRest/KSS Random Slope/DistUPvRest-RiSl")

DistUPvRest.Comparison = anova(DistUPvRest.ri$LM$Model, DistUPvRest.risl$LM$Model)
write(capture.output(DistUPvRest.Comparison), file = "./Processed
Data/StabDist/Models/Distinctiveness/UPvRest/UPvRest-ri-risl-comparison.txt")

# Distance of DNvRest -----
DistDNvRest.ri = Model_me_this(df.EEGBCI,
                                "`Distance Rest:DN` ~ Continuous.Block * Group + (1|PID)",
                                "./Processed
Data/StabDist/Models/Distinctiveness/DNvRest/Random Intercept/DistDNvRest-Ri")

DistDNvRest.ri.kss = Model_me_this(df.EEGBCI.KSS,
                                "`Distance Rest:DN` ~ Continuous.Block * Group + KSS +
(1|PID)",
                                "./Processed
Data/StabDist/Models/Distinctiveness/DNvRest/KSS Random Intercept/DistDNvRest-Ri")

DistDNvRest.risl = Model_me_this(df.EEGBCI,
                                "`Distance Rest:DN` ~ Continuous.Block * Group +
(Continuous.Block|PID)",
                                "./Processed
Data/StabDist/Models/Distinctiveness/DNvRest/Random Slope/DistDNvRest-RiSl")

DistDNvRest.risl.kss = Model_me_this(df.EEGBCI.KSS,
                                "`Distance Rest:DN` ~ Continuous.Block * Group + KSS +
(Continuous.Block|PID)",
                                "./Processed
Data/StabDist/Models/Distinctiveness/DNvRest/KSS Random Slope/DistDNvRest-RiSl")

```

```

DistDNvRest.Comparison = anova(DistDNvRest.ri$LM$Model, DistDNvRest.risl$LM$Model)
write(capture.output(DistDNvRest.Comparison), file = "./Processed
Data/StabDist/Models/Distinctiveness/DNvRest/DNvRest-ri-risl-comparison.txt")

# Investigate Obvious Interaction with Dist and something else -----
plot(df.EEGBCI$PID, df.EEGBCI$`Distance UP:DN`, col = ifelse(df.EEGBCI$Group == "TMS",
"blue", "green"))

# KVIQ may be up to something, this would also explain lack of effect!
plot(df.EEGBCI$`KVIQ Total`, df.EEGBCI$`Distance UP:DN`)
plot(df.EEGBCI$`KVIQ Imagery`, df.EEGBCI$`Distance UP:DN`)
plot(df.EEGBCI$`KVIQ Intensity`, df.EEGBCI$`Distance UP:DN`)

# Achievement measures? Maybe slope, lastSess, zero
plot(df.EEGBCI$t.Slope, df.EEGBCI$`Distance UP:DN`)
plot(df.EEGBCI$t.lastS, df.EEGBCI$`Distance UP:DN`)
plot(df.EEGBCI$t.zero, df.EEGBCI$`Distance UP:DN`)

# Less so other achievement measures
plot(df.EEGBCI$FirstVLast, df.EEGBCI$`Distance UP:DN`)
plot(df.EEGBCI$Achiever.kss, df.EEGBCI$`Distance UP:DN`)
plot(df.EEGBCI$Achiever.m, df.EEGBCI$`Distance UP:DN`)

# CA? doesn't seem to be it
plot(df.EEGBCI$CA, df.EEGBCI$`Distance UP:DN`, col = ifelse(df.EEGBCI$Sex == "Male",
"blue", "pink"))

# outliers? maybe
plot(df.EEGBCI$Continuous.Block, df.EEGBCI$`Distance UP:DN`, col =
ifelse(df.EEGBCI$Dist.UpDn.Outliers, "blue", "green"))
plot(df.EEGBCI$Continuous.Block, df.EEGBCI$`Distance UP:DN`, col =
ifelse(df.EEGBCI$Dist.Outliers, "blue", "green"))
plot(df.EEGBCI$`Distance UP:DN`[!df.EEGBCI$Dist.Outliers])

# Let's look at correlations:
library(psych)
pdf(file = "./Processed Data/StabDist/Models/Distinctiveness/UPvDN/Corrplot.pdf", width =
50, height = 50)
pairs.panels(df.EEGBCI,
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE, # show density plots
             ellipses = TRUE # show correlation ellipses
)
dev.off()

# Histogramm
plot(density(df.EEGBCI$`Distance UP:DN`))

# Separate distributions?
library(multimode)
modetest = modetest(df.EEGBCI$`Distance UP:DN`)
modetest
pdf(file = "./Processed Data/StabDist/Models/Distinctiveness/UPvDN/Distplot.pdf", width = 9,
height = 8)
Distance.distributions.UPvDN = locmodes(df.EEGBCI$`Distance UP:DN`,
mod0 = 3, display=TRUE)

dev.off()
Distance.distributions.UPvDN
write(capture.output(Distance.distributions.UPvDN), file = "./Processed
Data/StabDist/Models/Distinctiveness/UPvDN/ModeSep.txt")
write(capture.output(modetest), file = "./Processed
Data/StabDist/Models/Distinctiveness/UPvDN/Modetest.txt")

# split at antimode
df.EEGBCI$First.Distribution = df.EEGBCI$`Distance UP:DN` <=
Distance.distributions.UPvDN$locations[2]

```

```

df.EEGBCI$Second.Distribution = df.EEGBCI$`Distance UP:DN` >
Distance.distributions.UPvDN$locations[2]

df.EEGBCI.firstDist = df.EEGBCI[df.EEGBCI$First.Distribution == TRUE &
df.EEGBCI$Dist.Outliers == FALSE,]
df.EEGBCI.SecondDist = df.EEGBCI[df.EEGBCI$Second.Distribution == TRUE &
df.EEGBCI$Dist.Outliers == FALSE,]
df.EEGBCI.firstDist.kss = df.EEGBCI[df.EEGBCI$First.Distribution == TRUE &
df.EEGBCI$Dist.Outliers == FALSE &
!(is.na(df.EEGBCI$KSS)),]
df.EEGBCI.SecondDist.kss = df.EEGBCI[df.EEGBCI$Second.Distribution == TRUE &
df.EEGBCI$Dist.Outliers == FALSE &
!(is.na(df.EEGBCI$KSS)),]

# cull df for more managable things:
df.EEGBCI.firstDist.culled = df.EEGBCI.firstDist[, c("PID", "Continuous.Block", "Group",
"Distance UP:DN", "Distance Rest:UP",
"Distance Rest:DN", "Stability of
Rest", "Stability of UP", "Stability of DN", "Stab.Outliers", "Dist.Outliers",
"KSS", "CA", "Success Cross",
"Success Hand", "Failure Bias", "Sex", "Age", "KVIQ Total", "Achiever.kss", "Achiever.m",
"t.zero",
"t.lastS", "t.Slope", "FirstVLast")]

library(psych)
pdf(file = "../Processed Data/StabDist/Models/Distinctiveness/UPvDN/Corrplot-FirstDist.pdf",
width = 50, height = 50)
pairs.panels(df.EEGBCI.firstDist.culled,
method = "pearson", # correlation method
hist.col = "#00AFBB",
density = TRUE, # show density plots
ellipses = TRUE, # show correlation ellipses
stars = TRUE
)
dev.off()
pdf(file = "../Processed Data/StabDist/Models/Distinctiveness/UPvDN/Corrplot-culled-
DistNoOut.pdf", width = 50, height = 50)
pairs.panels(df.EEGBCI[df.EEGBCI$Dist.Outliers == FALSE, c("PID", "Continuous.Block",
"Group", "Distance UP:DN", "Distance Rest:UP",
"Distance Rest:DN", "Stability
of Rest", "Stability of UP", "Stability of DN", "KSS", "CA",
"Success Cross", "Success
Hand", "Failure Bias", "Sex", "Age", "KVIQ Total", "Achiever.kss",
"Achiever.m", "t.zero",
"t.lastS", "t.Slope", "FirstVLast", "First.Distribution", "Second.Distribution")],
method = "pearson", # correlation method
hist.col = "#00AFBB",
density = TRUE, # show density plots
ellipses = TRUE, # show correlation ellipses
stars = TRUE
)
dev.off()

pdf(file = "../Processed Data/StabDist/Models/Distinctiveness/UPvDN/Corrplot-SecDist.pdf",
width = 50, height = 50)
pairs.panels(df.EEGBCI.SecondDist[df.EEGBCI.SecondDist$Dist.Outliers == FALSE, c("PID",
"Continuous.Block", "Group", "Distance UP:DN", "Distance Rest:UP",
"Distance
Rest:DN", "Stability of Rest", "Stability of UP", "Stability of DN", "KSS", "CA",
"Success
Cross", "Success Hand", "Failure Bias", "Sex", "Age", "KVIQ Total", "Achiever.kss",
"Achiever.m", "t.zero", "t.lastS", "t.Slope", "FirstVLast", "First.Distribution",
"Second.Distribution")],
method = "pearson", # correlation method
hist.col = "#00AFBB",
density = TRUE, # show density plots
ellipses = TRUE, # show correlation ellipses
stars = TRUE
)

```

```

)
dev.off()

# Let's look at the dist:
plot(df.EEGBCI$CA, df.EEGBCI$`Distance UP:DN`, col = ifelse(df.EEGBCI$Group == "TMS",
"blue", "green"))

# Ok without Outliers
df.EEGBCI.distnoOut = df.EEGBCI[!df.EEGBCI$Dist.Outliers, ]
plot(df.EEGBCI$CA,
df.EEGBCI$`Distance UP:DN`,
col = ifelse(df.EEGBCI$Group == "TMS", "blue", "green"))

# Distance UPvDN split -----

DistUpvDn.split.ri = Model_me_this(df.EEGBCI.firstDist,
                                "`Distance UP:DN` ~ Continuous.Block * Group + (1|PID)",
                                "./Processed
Data/StabDist/Models/Distinctiveness/UPvDN/First Dist Random Intercept/FirstDist-DistUpvDn-
Ri")

DistUpvDn.split.risl = Model_me_this(df.EEGBCI.firstDist,
                                "`Distance UP:DN` ~ Continuous.Block * Group +
(Continuous.Block|PID)",
                                "./Processed
Data/StabDist/Models/Distinctiveness/UPvDN/First Dist Random Slope/FirstDist-DistUpvDn-
RiSl")

DistUpvDn.split.kss.ri = Model_me_this(df.EEGBCI.firstDist.kss,
                                "`Distance UP:DN` ~ Continuous.Block * Group + KSS +
(1|PID)",
                                "./Processed
Data/StabDist/Models/Distinctiveness/UPvDN/KSS First Dist Random Intercept/KSSFirstDist-
DistUpvDn-Ri")

DistUpvDn.split.kss.risl = Model_me_this(df.EEGBCI.firstDist.kss,
                                "`Distance UP:DN` ~ Continuous.Block * Group + KSS
+ (Continuous.Block|PID)",
                                "./Processed
Data/StabDist/Models/Distinctiveness/UPvDN/KSS First Dist Random Slope/KSSFirstDist-
DistUpvDn-RiSl")

# None of them really interesting to be honest
UPvDN = list()
UPvDN$Ind.tTest = IndependantTest(df.EEGBCI$`Distance UP:DN`[df.EEGBCI$Group == "TMS"],
                                df.EEGBCI$`Distance UP:DN`[df.EEGBCI$Group == "EEG"],
                                testName = "t.test Distance UPvDN Groupwise",
                                directionality = "g")
write(capture.output(UPvDN$Ind.tTest), file = "./Processed
Data/StabDist/Models/Distinctiveness/UPvDN/IndtTest.txt")

UPvDN$Ind.tTest.FirstDist = IndependantTest(df.EEGBCI.firstDist$`Distance
UP:DN`[df.EEGBCI$Group == "TMS"],
                                df.EEGBCI.firstDist$`Distance
UP:DN`[df.EEGBCI$Group == "EEG"],
                                testName = "t.test Distance UPvDN Groupwise,
first Distribution",
                                directionality = "g")
write(capture.output(UPvDN$Ind.tTest.FirstDist), file = "./Processed
Data/StabDist/Models/Distinctiveness/UPvDN/IndtTest-FirstDist.txt")

UPvDN$Ind.tTest.SecondDist = IndependantTest(df.EEGBCI.SecondDist$`Distance
UP:DN`[df.EEGBCI$Group == "TMS"],
                                df.EEGBCI.SecondDist$`Distance
UP:DN`[df.EEGBCI$Group == "EEG"],
                                testName = "t.test Distance UPvDN Groupwise,
Second Distribution",
                                directionality = "g")

```

```

write(capture.output(UPvDN$Ind.tTest.SecondDist), file = "../Processed
Data/StabDist/Models/Distinctiveness/UPvDN/IndtTest-SecondDist.txt")

# Reintegration into CA -----
# effect of UPvDN distance on CA: (don't expect mjch here)
UPvDN.CA = list()
UPvDN.CA$Basic.Model = Model_me_this(df.EEGBCI,
                                     "CA ~ Continuous.Block * Group + `Distance UP:DN` +
(1|PID)",
                                     "../Processed
Data/StabDist/Models/Distinctiveness/UPvDN/UPvDN-CA/UPvDN-CA")
UPvDN.CA$Basic.Model.risl = Model_me_this(df.EEGBCI,
                                     "CA ~ Continuous.Block * Group + `Distance UP:DN` +
(Continuous.Block|PID)",
                                     "../Processed
Data/StabDist/Models/Distinctiveness/UPvDN/UPvDN-CA/RiSl-UPvDN-CA")
UPvDN.CA$Basic.Model.KSS = Model_me_this(df.EEGBCI.KSS,
                                     "CA ~ Continuous.Block * Group + KSS + `Distance
UP:DN` + (1|PID)",
                                     "../Processed
Data/StabDist/Models/Distinctiveness/UPvDN/UPvDN-int-CA/KSS-UPvDN-CA")
UPvDN.CA$Int.Model = Model_me_this(df.EEGBCI,
                                     "CA ~ Continuous.Block * Group * `Distance UP:DN` +
(1|PID)",
                                     "../Processed
Data/StabDist/Models/Distinctiveness/UPvDN/UPvDN-CA/UPvDN-int-CA")
UPvDN.CA$Int.Model.risl = Model_me_this(df.EEGBCI,
                                     "CA ~ Continuous.Block * Group * `Distance UP:DN`
+ (Continuous.Block|PID)",
                                     "../Processed
Data/StabDist/Models/Distinctiveness/UPvDN/UPvDN-int-CA/RiSl-UPvDN-int-CA")
UPvDN.CA$Int.Model.KSS = Model_me_this(df.EEGBCI.KSS,
                                     "CA ~ Continuous.Block * Group * `Distance UP:DN`
+ KSS + (1|PID)",
                                     "../Processed
Data/StabDist/Models/Distinctiveness/UPvDN/UPvDN-int-CA/KSS-UPvDN-int-CA")
UPvDN.CA$Theory = Model_me_this(df.EEGBCI,
                                     "CA ~ Continuous.Block * Group +
Continuous.Block:Group:`Distance UP:DN` + (1|PID)",
                                     "../Processed
Data/StabDist/Models/Distinctiveness/UPvDN/UPvDN-CA/UPvDN-theory-CA")

# Last Effort - DistRestUp because Balanced! -----
Dist.RestUP.ri = Model_me_this(df.EEGBCI,
                               "CA ~ Continuous.Block * Group + `Distance Rest:UP` +
(1|PID)",
                               "../Processed
Data/StabDist/Models/Distinctiveness/UPvRest/CA-Basic/CA-DistUpvDn-Ri")
Dist.RestUP.risl = Model_me_this(df.EEGBCI,
                               "CA ~ Continuous.Block * Group + `Distance Rest:UP` +
(Continuous.Block|PID)",
                               "../Processed
Data/StabDist/Models/Distinctiveness/UPvRest/CA-RiSl/CA-DistUpvDn-RiSl")
dist.RestUP.risl.comparison = anova(Dist.RestUP.ri$LM$Model, Dist.RestUP.risl$LM$Model)
write(capture.output(dist.RestUP.risl.comparison), file = "../Processed
Data/StabDist/Models/Distinctiveness/UPvRest/Risl-Comparison.txt")

# Ri is as good, proceed with that
Dist.RestUP.ri.kss = Model_me_this(df.EEGBCI.KSS,
                                   "CA ~ Continuous.Block * Group + `Distance Rest:UP` + KSS +
(1|PID)",
                                   "../Processed
Data/StabDist/Models/Distinctiveness/UPvRest/CA-KSS/CA-KSS-DistUpvDn-Ri")
Dist.RestUP.risl.kss = Model_me_this(df.EEGBCI.KSS,

```



```

"CA ~ Continuous.Block * Group + `Distance Rest:UP` + KSS +
(Continuous.Block|PID)",
    "./Processed
Data/StabDist/Models/Distinctiveness/UPvRest/CA-KSS/CA-KSS-DistUpvDn-RiSl")
dist.RestUP.risl.kss.comparison = anova(Dist.RestUP.ri.kss$LM$Model,
Dist.RestUP.risl.kss$LM$Model)
write(capture.output(dist.RestUP.risl.kss.comparison), file = "./Processed
Data/StabDist/Models/Distinctiveness/UPvRest/KSS-RiSl-Comparison.txt")

# Interesting. UP seems to have an effect. Need to look at interactions.
Dist.RestUP.ri.int = Model_me_this(df.EEGBCI,
    "CA ~ Continuous.Block * Group + `Distance Rest:UP` +
Group:`Distance Rest:UP` + (1|PID)",
    "./Processed
Data/StabDist/Models/Distinctiveness/UPvRest/CA-Int/CA-DistUpvDn-Ri")
Dist.RestUP.risl.int = Model_me_this(df.EEGBCI,
    "CA ~ Continuous.Block * Group + `Distance Rest:UP` +
Continuous.Block:`Distance Rest:UP` + (Continuous.Block|PID)",
    "./Processed
Data/StabDist/Models/Distinctiveness/UPvRest/CA-Int/CA-DistUpvDn-RiSl")
dist.RestUP.risl.comparison = anova(Dist.RestUP.ri.int$LM$Model,
Dist.RestUP.risl.int$LM$Model)
write(capture.output(dist.RestUP.risl.comparison), file = "./Processed
Data/StabDist/Models/Distinctiveness/UPvRest/Int-Comparison.txt")

# Last Effort - RestDN -----
Dist.RestDN.ri = Model_me_this(df.EEGBCI,
    "CA ~ Continuous.Block * Group + `Distance Rest:DN` +
(1|PID)",
    "./Processed
Data/StabDist/Models/Distinctiveness/DNvRest/CA-Basic/CA-DistUpvDn-Ri")
Dist.RestDN.risl = Model_me_this(df.EEGBCI,
    "CA ~ Continuous.Block * Group + `Distance Rest:DN` +
(Continuous.Block|PID)",
    "./Processed
Data/StabDist/Models/Distinctiveness/DNvRest/CA-Basic/CA-DistUpvDn-RiSl")
dist.RestDN.risl.comparison = anova(Dist.RestDN.ri$LM$Model, Dist.RestDN.risl$LM$Model)
write(capture.output(dist.RestDN.risl.comparison), file = "./Processed
Data/StabDist/Models/Distinctiveness/DNvRest/Risl-Comparison.txt")

# Ri is as good, proceed with that
Dist.RestDN.ri.kss = Model_me_this(df.EEGBCI.KSS,
    "CA ~ Continuous.Block * Group + `Distance Rest:DN` +
KSS + (1|PID)",
    "./Processed
Data/StabDist/Models/Distinctiveness/DNvRest/CA-KSS/CA-KSS-DistUpvDn-Ri")
Dist.RestDN.risl.kss = Model_me_this(df.EEGBCI.KSS,
    "CA ~ Continuous.Block * Group + `Distance Rest:DN` +
KSS + (Continuous.Block|PID)",
    "./Processed
Data/StabDist/Models/Distinctiveness/DNvRest/CA-KSS/CA-KSS-DistUpvDn-RiSl")
dist.RestDN.risl.kss.comparison = anova(Dist.RestDN.ri.kss$LM$Model,
Dist.RestDN.risl.kss$LM$Model)
write(capture.output(dist.RestDN.risl.kss.comparison), file = "./Processed
Data/StabDist/Models/Distinctiveness/DNvRest/KSS-RiSl-Comparison.txt")

# Interesting. DN seems to have an effect. Need to look at interactions.
Dist.RestDN.ri.int = Model_me_this(df.EEGBCI,
    "CA ~ Continuous.Block * Group + Group:`Distance
Rest:DN` + (1|PID)",
    "./Processed
Data/StabDist/Models/Distinctiveness/DNvRest/CA-Int/CA-DistUpvDn-Ri")
Dist.RestDN.risl.int = Model_me_this(df.EEGBCI,
    "CA ~ Continuous.Block * Group +
Continuous.Block:`Distance Rest:DN` + (Continuous.Block|PID)",
    "./Processed
Data/StabDist/Models/Distinctiveness/DNvRest/CA-Int/CA-DistUpvDn-RiSl")

```



```

dist.RestDN.risl.comparison = anova(Dist.RestDN.risl.int$LM$Model,
Dist.RestDN.risl.int$LM$Model)
write(capture.output(dist.RestDN.risl.comparison), file = "./Processed
Data/StabDist/Models/Distinctiveness/DNvRest/Int-Comparison.txt")

# A bit more UPvDN -----
upvdn.m1 = Model_me_this(df.EEGBCI.firstDist.culled,
  "CA ~ Continuous.Block * Group + `Distance UP:DN` + (1|PID)",
  "./Processed Data/StabDist/Models/Distinctiveness/UPvDN/CA-FirstDist/CA-
FirstDist-UpvDn-Ri")
upvdn.m2 = Model_me_this(df.EEGBCI.firstDist.culled,
  "CA ~ Continuous.Block * Group + `Distance UP:DN` + `Distance UP:DN`:Group +
(1|PID)",
  "./Processed Data/StabDist/Models/Distinctiveness/UPvDN/CA-FirstDist-
Group/CA-FirstDist-UpvDn-Ri")
upvdn.m3 = Model_me_this(df.EEGBCI.firstDist.culled,
  "CA ~ Continuous.Block * Group + `Distance UP:DN` + `Distance
UP:DN`:Continuous.Block + (1|PID)",
  "./Processed Data/StabDist/Models/Distinctiveness/UPvDN/CA-FirstDist-
Block/CA-FirstDist-UpvDn-Ri")
upvdn.m4 = Model_me_this(df.EEGBCI.firstDist.culled,
  "CA ~ Continuous.Block * Group * `Distance UP:DN` + (1|PID)",
  "./Processed Data/StabDist/Models/Distinctiveness/UPvDN/CA-FirstDist-comp/CA-
FirstDist-UpvDn-Ri")
upvdn.comp = anova(upvdn.m1$LM$Model,
  upvdn.m2$LM$Model,
  upvdn.m3$LM$Model,
  upvdn.m4$LM$Model)

Model_me_this(df.EEGBCI.firstDist.culled,
  "CA ~ Continuous.Block * Group * `Distance Rest:DN` + (1|PID)",
  "./Processed Data/StabDist/Models/Distinctiveness/DNvRest/CA-FirstDist/CA-
FirstDist-compint-RestvDn-Ri")
Model_me_this(df.EEGBCI.firstDist.culled,
  "CA ~ Continuous.Block * Group * `Distance Rest:UP` + (1|PID)",
  "./Processed Data/StabDist/Models/Distinctiveness/UPvRest/CA-FirstDist/CA-
FirstDist-compint-UpvRest-Ri")

Model_me_this(df.EEGBCI.firstDist.culled[!is.na(df.EEGBCI.firstDist.culled$KSS), ],
  "CA ~ Continuous.Block * Group + `Distance UP:DN` + KSS + (1|PID)",
  "./Processed Data/StabDist/Models/Distinctiveness/UPvDN/CA-FirstDist-KSS/CA-
FirstDist-KSS-UpvDn-Ri")

Model_me_this(df.EEGBCI.firstDist.culled[!is.na(df.EEGBCI.firstDist.culled$KSS), ],
  "CA ~ Continuous.Block * Group + `Distance UP:DN` + KSS +
(Continuous.Block|PID)",
  "./Processed Data/StabDist/Models/Distinctiveness/UPvDN/CA-FirstDist-KSS-
RiSl/CA-FirstDist-KSS-RiSl-UpvDn-Ri")

Model_me_this(df.EEGBCI.firstDist.culled[!is.na(df.EEGBCI.firstDist.culled$KSS), ],
  "CA ~ Continuous.Block * Group + `Distance UP:DN` + Group:`Distance UP:DN` +
KSS + (1|PID)",
  "./Processed Data/StabDist/Models/Distinctiveness/UPvDN/CA-FirstDist-KSS-
Group/CA-FirstDist-KSS-Group-UpvDn-Ri")

Model_me_this(df.EEGBCI.firstDist.culled[!is.na(df.EEGBCI.firstDist.culled$KSS), ],
  "CA ~ Continuous.Block * Group + `Distance UP:DN` + Group:`Distance
UP:DN`:Continuous.Block + KSS + (1|PID)",
  "./Processed Data/StabDist/Models/Distinctiveness/UPvDN/CA-FirstDist-KSS-
Group-comp/CA-FirstDist-KSS-Group-comp-UpvDn-Ri")

Model_me_this(df.EEGBCI.firstDist.culled[!is.na(df.EEGBCI.firstDist.culled$KSS), ],
  "CA ~ Continuous.Block * Group + `Distance UP:DN` * KSS + (1|PID)",
  "./Processed Data/StabDist/Models/Distinctiveness/UPvDN/CA-FirstDist-KSS-
comp/CA-FirstDist-KSS-comp-UpvDn-Ri")

Model_me_this(df.EEGBCI.firstDist.culled,

```

```

      "CA ~ Continuous.Block * Group + Continuous.Block:`Distance UP:DN` +
(1|PID)",
      "./Processed Data/StabDist/Models/Distinctiveness/UPvDN/CA-FirstDist-comp/CA-
FirstDist-comp-KSS-UpvDn-Ri")

Model_me_this(df.EEGBCI.firstDist.culled[!is.na(df.EEGBCI.firstDist.culled$KSS), ],
      "CA ~ Continuous.Block * Group * `Distance UP:DN` + KSS + (1|PID)",
      "./Processed Data/StabDist/Models/Distinctiveness/UPvDN/CA-FirstDist-KSS-
compint/CA-FirstDist-KSS-Group-comp-UpvDn-Ri")

```