Measuring Dynamical Shifts Following Transcranial Magnetic Stimulation

By Rishabh Bapat

A dissertation submitted to the faculty of the National Brain Research Centre
in Partial Fulfillment of the Requirements for a Master of Science in Neuroscience.
On June 19, 2023

National Brain Research Centre
NBRC

DECLARATION BY CANDIDATE

I, **Rishabh Bapat** hereby declare that the work presented in the dissertation report entitled **‘Measuring Dynamical Shifts Following Transcranial Magnetic Stimulation’** was carried out by me under the guidance of **Dr. Arpan Banerjee**. In the division of Cognitive Brain dynamics. **National Brain Research Centre (Deemed to be University),** Manesar, Haryana, India. I declare that no part of the thesis contains any plagiarised material. Any previously published or other material sourced from anywhere else has been appropriately attributed to the source. I also declare that no part of this thesis has been previously submitted for the award of any degree of diploma to **National Brain Research Centre (Deemed to be University)** or to any other university.

Place : Manesar (Name and Signature of the candidate)

Date :

CERTIFICATE

This is to certify that the dissertation report entitled **‘Measuring Dynamical Shifts Following Transcranial Magnetic Stimulation’** is the result of work carried out by **Rishabh Bapat** in the Division of **Cognitive Brain Dynamics**, National Brain Research Centre, Manesar, Haryana, India, for partial fulfilment of requirements for the M.Sc. degree.

The work presented herein is original and has not been submitted previously for the award of any degree or diploma to **National Brain Research Centre (Deemed to be University)** or to any other University. This work is based on the guidelines of **National Brain Research Centre (Deemed to be University)** and is a record of the candidate’s own efforts.

(Supervisor’s name and signature) (Director, NBRC)

Place : Manesar

Date :

Acknowledgements

This project has been a labor of love, a learning experience and a series of hurdles at the same time. Throughout the process I have accrued a great many debts of gratitude that I shall now acknowledge. Firstly, I would like to thank my supervisor Dr. Arpan Banerjee, whose constant guidance and assistance have been indispensable throughout this project. The free reign he gave me over my project allowed me to express my passion for neuroscience, and his patience made me unafraid to try new things in pursuit of knowledge. Another debt is owed to my friends and colleagues, Ajay, Mihir, Shivam, Tanmay and Tulika, who shared my journey and filled me with spirit after success and failure alike. I would also like to thank my family, whose constant love and support have made not just this project, but my entire career in neuroscience possible.

Next I wish to appreciate all the faculty and most of the staff at the National Brain Research Centre for their support. Each faculty has taught me invaluable lessons, both in neuroscience and in life, which will remain with me throughout my career. I would be remiss, If I didn't mention the countless insights provided to me by the members of the Cognitive Brain Dynamics Lab. Their patient guidance allowed me to surmount many obstacles and this work would certainly not be possible without them. Lastly, I would like to note that no list of acknowledgements could summarize the myriad sources of assistance I have been fortunate enough to receive. Therefore this list is by no means exhaustive and I hold in my heart the contributions of countless others that must go unmentioned.

Thanking you,
Rishabh Bapat

Abstract

The human brain is composed of modular neural ensembles, each one efficient at an elementary function. Neural coordination among these ensembles occurs flexibly to perform complex tasks and reorganizes dynamically over time. Recent studies posit that underlying this ability to switch between coordination states, is the tendency of the system to move between periods of stability and instability, quantitatively captured by a mathematical measure, metastability. Previous literature has shown that metastable dynamics are important for optimal functioning, and that metastability is disordered in several pathologies. Despite its crucial place in coordination dynamics, metastability and how it changes in response to brain stimulation is an unexplored area. In this study, changes in metastability in response to Transcranial Magnetic Stimulation (TMS) were investigated. TMS-EEG data acquired from OpenNeuro were analyzed with two measures of metastability, one based on the variability in coherence, and the other based on the entropy of spatiotemporal dynamics. We observed a significant reduction in metastability, concurrent with an increase in coherence. Additionally, higher frequencies showed faster recovery in metastability than lower frequencies. These results are in line with theoretical expectations and validate the measures applied. Interestingly an increase in metastability prior to the pulse is also observed in the alpha and theta bands, given that a stimulus preceding negativity was also observed prior to the pulse, these results suggest a relationship between anticipation and increased metastability. Overall, this study shows that TMS reliably perturbs metastability, suggesting the use of TMS to investigate the functional role of metastability. Conversely the dynamic measures of metastability applied can be used to index the effects of various TMS protocols.

Contents

Table of Abbreviations

Abbreviation	**Meaning**
EEG	Electroencephalogram
TMS	Transcranial Magnetic Stimulation
spTMS	Single Pulse Transcranial Magnetic Stimulation
RMT	Resting Motor Threshold
rPMC	Right Primary Motor Cortex
KOP	Kuramoto Order Parameter
LZC	Lempel-Ziv Complexity
GFP	Global Field Power
ERP	Event Related Potential
SPN	Stimulus Preceding Negativity

Chapter 1

Introduction

1.1 The Brain as a System of Oscillators

The brain is the electrochemical substrate of our very being. It is an intricate and beautiful system wherein a myriad of deterministic elements collectively ascend to consciousness. Since the earliest attempts to measure the electrical dynamics underlying its operation, one thing has been evident; Neural dynamics are oscillatory (Berger, 1929).

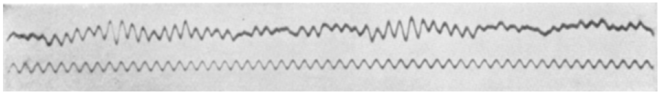

Figure 1.1: An early EEG recording by Hans Berger.

Neural potentials generally oscillate between 1 and 100 hertz and the frequencies that tend to change together have been grouped into ranges as follows,

1. Delta (1 to 4 Hz), occurs during deep sleep.
2. Theta (4 to 7 Hz), involved in memory consolidation.
3. Alpha (7 to 12 Hz), involved in the gating of sensory information.
4. Beta (13 to 30 Hz), involved in motor coordination.
5. Gamma (30 to 70 Hz), involved in coordinating activity across sensory regions

These oscillations are thought to emerge due to interactions at various scales. At the microscopic scale, the periodic nature of the action potential, and the sub threshold oscillations observed in certain neurons contribute to this effect. At the mesoscopic scale it is the synchronisation of neural ensembles and interactions between excitatory and inhibitory populations.

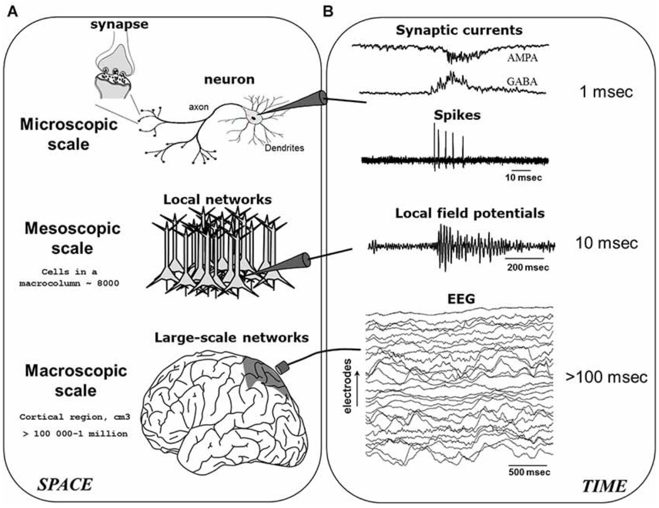

Figure 1.2: The origin of neural oscillations (Le Van Quyen, 2011)

Finally, at the macroscopic scale feedback interactions between cortical areas produce oscillatory patterns of activity (Cohen, 2014).

Given their ubiquity in neural dynamics, oscillations present an interesting lens by which to analyse the dynamics of the brain. One perspective treats the brain as a system of interacting oscillators, that encode information into changes in frequency, amplitude and phase. The factor controlling the flow of information thus encoded, is the coherence between oscillating populations.

The dynamics of coherence across the brain can thus represent how information is flowing across the cortex. In recent years an investigation of said dynamics has revealed an interesting property of the brain, metastability.

1.2 Metastability

In simple oscillatory systems, the dynamics converge to complete synchrony if the coupling is strong enough, or remain asynchronous if the coupling is too weak (Cabral et al., 2011). In these situations a single stable state exists and the system is confined by it. However, dynamics in the brain manifest differently; the system moves between a range of partially coherent states, only briefly dwelling at each one before transitioning to the next. These dynamics in the absence of an attractor or noise are defined as metastability (Tognoli & Kelso, 2014). The coherence between regions changing in this way has several advantages,

1. Regions with heterogeneous intrinsic dynamics can be coupled and exchange information.

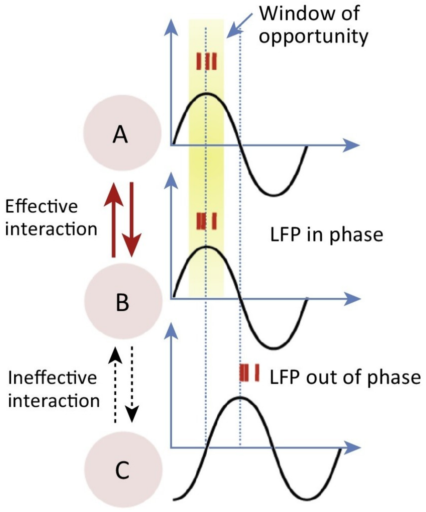

Figure 1.3: How coherence affects information transfer (Deco & Kringelbach, 2016)

2. The system is able to switch between coupling states without becoming confined to them.
3. No energy input is required to transition the system between coupling states.

As a consequence of this, metastability enables the brain to fully explore its dynamical potential. Reflecting the fundamental role of metastability, modeling studies have found it to be critical to the emergence of coupling through coherence (Deco & Kringelbach, 2016) and maximized in the resting state (Deco et al., 2017; Hellyer et al., 2014). A plethora of studies further validate this claim by showing changes in metastability to accompany altered or disordered states of consciousness,

- Large reductions in metastability (during epilepsy or sleep) cause loss of consciousness (Cavanna et al., 2018; Jobst et al., 2017).
- Metastability is reduced following traumatic brain injury (Hellyer et al., 2015).
- Progressive reduction observed in Alzheimer's disease (Córdova-Palomera et al., 2017).
- Metastability is found to be higher among schizophrenics (Lee et al., 2018).
- Metastability increases following the use of psychedelic drugs (Carhart-Harris et al., 2014; Lord et al., 2019).

Some of these studies also indicate changes in cognitive flexibility caused by reduction in metastability (Córdova-Palomera et al., 2017; Hellyer et al., 2015). In summary, when viewed as a system of oscillators, the brain exhibits a property known as metastability which is crucial for its operation. Changes to it manifest in altered consciousness and functioning.

1.3 Transcranial Magnetic Stimulation

In Transcranial Magnetic Stimulation (TMS), an electromagnet is used to briefly create a magnetic field over a part of the cortex, thereby inducing an electrical current. This method is used to non invasively perturb the cortex and can be used to excite or inhibit a region depending on the stimulation protocol and location (Klomjai et al., 2015). TMS protocols can be classified based on the frequency of stimulation as follows,

1. Single pulse: A single monophasic, or biphasic magnetic pulse is delivered
2. Paired pulse: Two pulses are delivered to separate areas within a short delay.
3. Repetitive: Pulses are delivered continuously at a certain frequency.
4. Burst: Pulses are delivered in high frequency bursts intermittently.

Single pulse TMS has been used in research contexts to interfere with cognition (Robertson et al., 2003) and in clinical settings to test the integrity of motor pathways (Perez & Cohen, 2009). Repetitive protocols and burst stimulation can increase the excitability of the stimulated region or increase the connectivity between two regions. This is being used as a treatment for drug resistant depression (Eshel et al., 2020). Despite the wide usage of TMS in both research and clinical settings, there exists tremendous variability in its effects (Corp et al., 2021). An index of neural dynamics that characterises the effects of a given protocol would be extremely useful to help optimise protocols and replicate results. TMS is known to cause phase locking in the stimulated region. This would effectively force the brain into a specific coupling state, altering metastability transiently. Paired and repetitive TMS are also known to produce long term effects on coherence and excitability which could produce longer lasting changes to metastability. Thus metastability could be used as an index to describe the effects of a given TMS protocol. It has utility over simpler spectral measures, since it is preserved across timescales, allowing it to describe both transient and long term effects of TMS. Another advantage is that it relates more directly to higher order functions, and its derangement during disorders of consciousness is documented. Hence optimising clinical protocols with respect to it could enhance their effectiveness. Finally studying how metastability changes in response to TMS would help explain the mechanism by which certain protocols have their effects and in turn better describe how metastability relates to cognitive functioning.

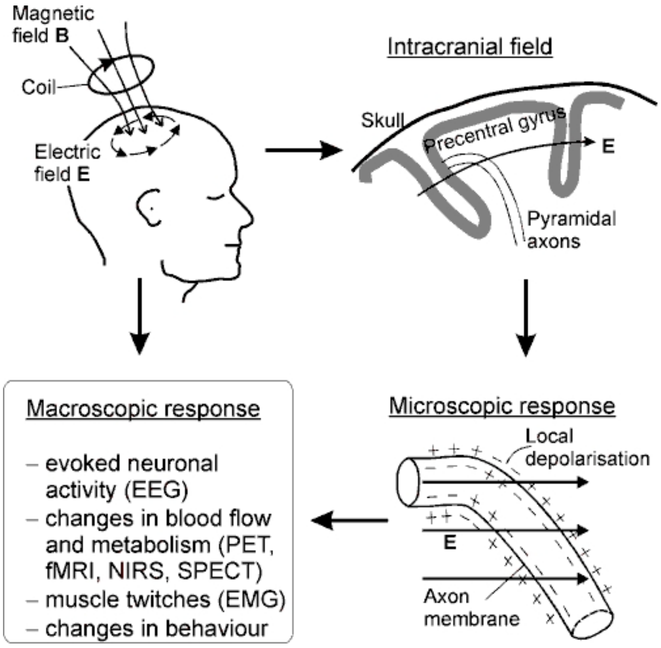

Figure 1.4: Working principle of TMS (Viesca et al., 2012)

1.4 Goals

The goals of this dissertation are as follows,

1. To investigate how metastability changes immediately after single pulse monophasic TMS.

2. To establish whether metastability can be used to index the immediate effects of TMS protocols.

3. To use the nature of this effect to improve our understanding of how metastability relates to cognition.

Chapter 2

Methods

2.1 Data Collection

The data used in this study was obtained from OpenNeuro. One dataset had 20 participants (Hussain, 2019) while the other had 13 (Pavon et al., 2022). 300 seconds of resting state, eyes open EEG were recorded for estimation of Resting Motor Threshold (RMT) using either a 30 channel electrode array and a 5kHz sampling rate (Hussain, 2019) or a 63 channel electrode array with a 20kHz sampling rate (Pavon et al., 2022). Single pulse, monophasic TMS was delivered to the right Primary Motor Cortex (rPMC) at 100%, 110% and 120% of RMT every 5 seconds. 600 trials at 120% of RMT were conducted in one dataset (Hussain, 2019) with 75 trials at 110, 100% of RMT in the other (Pavon et al., 2022). EEG recordings were taken simultaneously. The coil noise was not masked, but participants were provided with earplugs to reduce the disturbance.

2.2 Preprocessing

The resting state and TMS stimulated data were preprocessed prior to analysis. The resting state data were processed using custom MATLAB code utilising a combination of Artifact Subspace Reconstruction (Plechawska-Wojcik et al., 2019) and MARA (Winkler et al., 2011) based independent component rejection. The TMS stimulated data were preprocessed using the ARTIST pipeline (Wu et al., 2018). All data was downsampled to 1kHz and bandpass filtered between 1 and 100Hz prior to analysis.

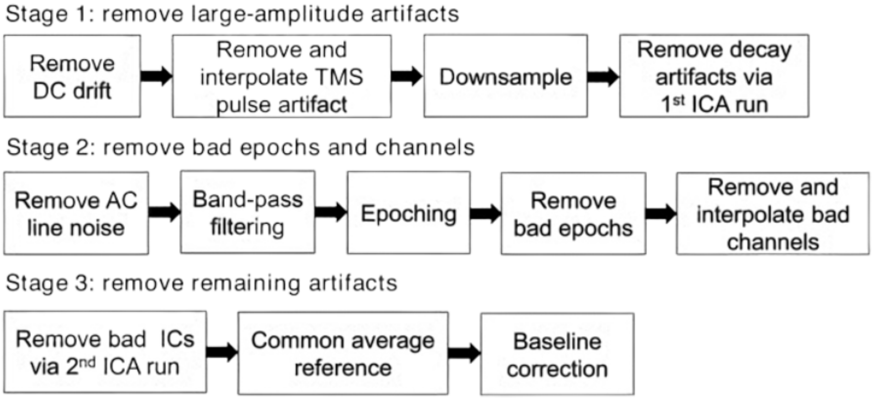
Stage 1: remove large-amplitude artifacts
Remove DC drift
Remove and interpolate TMS pulse artifact
Downsample
Remove decay artifacts via 1st ICA run
Stage 2: remove bad epochs and channels
Remove AC line noise
Band-pass filtering
Epoching
Remove bad epochs
Remove and interpolate bad channels
Stage 3: remove remaining artifacts
Remove bad ICs via 2nd ICA run
Common average reference
Baseline correction

Figure 2.1: ARTIST pipeline (Wu et al., 2018)

2.3 Measures of Metastability

The Standard Deviation of the Kuramoto Order Parameter

Two measures of metastability were then applied to the preprocessed data. The first measure was based on the standard deviation of the Kuramoto Order Parameter (KOP) based on the work of Wildie and Shanahan (2012). The Kuramoto order parameter is a measure of synchrony in a network of oscillators. It is calculated by using the following formula,

$$re^{i\psi} = \frac{1}{N}\sum_{j=1}^{N} e^{i\theta_j}$$

Where 'N' is the number of oscillators and 'θ_j' is the phase of the jth oscillator and 'r' is the Kuramoto order parameter. The KOP can be thought of as plotting the phase of each oscillator as a point on a unit circle, and then taking the magnitude of their resultant vector. Its value is 1 for a completely synchronised system and 0 for a desynchronised system. Its standard deviation indicates the ability of the system to deviate from stable states and thus can be used as a measure of metastability. For this measure, the TMS EEG data were bandpass filtered within narrow frequency bands (for example, 8 to 12 Hz), after which the instantaneous phase was extracted via Hilbert transformation. The KOP was then calculated for each time point. Its standard deviation in a sliding window of 50 ms was used as a dynamic measure of metastability. The metastability timeseries were then averaged across epochs and participants to yield the final results.

The choice of which group of channels to include in computing this measure is an important one. Since changes to global coherence (all channels) are relatively less intense, and stimulation with TMS has prominent local effects, channel groupings derived from an algorithm were analysed in addition to global metastability. This algorithm aimed to capture local effects on synchrony and grouped channels together based on the similarity of their TMS evoked potential. First the evoked response to TMS would be calculated in each stimulation intensity by averaging across epochs and subjects. Then the timepoint with the highest Global Field Power (GFP), meaning the highest spatial standard deviation, was selected. At this time point, the activity of each channel was subtracted from the mean of all channels. The channels were then sorted in ascending order of their mean differences, and the top and bottom tertiles were labeled as group 1 and group 2 respectively. This algorithm is analogous to the process of visually inspecting the evoked response, finding the point at which the channels activity is the most varied, and sorting the most extreme channels into their own groups.

Microstate Sequence Complexity

The second measure was devised for this study and is based on microstate sequence complexity. Microstates are quasi-stable spatial activity patterns that are derived from EEG data using clustering algorithms. The original recording can then be backfitted to the most similar microstate at each point in time and be analysed as a sequence of microstates (Lehmann, 1971). The quasi-stable nature of these states is reminiscent of the dwell and escape tendencies seen in metastable dynamics while the microstates themselves are related to the coordination states whose metastability is the subject of this study. The central idea behind this measure is that metastable coordination dynamics will produce a non-repetitive microstate sequence. This can be quantified for the microstate sequence using measures of ‘complexity’.

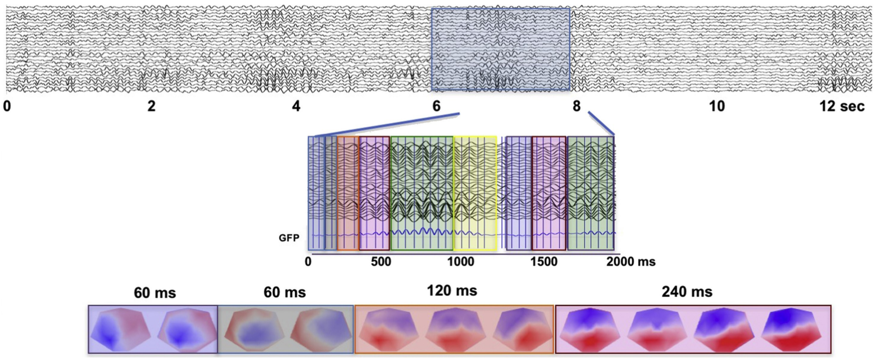

Figure 2.2: Overview of microstate analysis (Michel & Koenig, 2018)

‘Complexity’ refers to the unpredictability of a signal and is quantified here by Lempel-Ziv Complexity (LZC). The LZC of a string is the minimum number of unique sub-strings that can be repeated and combined to reproduce the original. It increases with the unpredictability and length of a string (Lempel & Ziv, 1986). In order to compute this measure, microstates were derived from GFP peaks in resting state data that was bandpassed to alpha (8-12 Hz). The clustering was done using Topographical Atomise and Agglomerate Hierarchical Clustering. In this method initially, each timepoint is its own microstate. Through iteratively eliminating (atomising) and redistributing (agglomerating) the worst microstate, based on the sum of correlations between the microstate and its members (Correlation Sum), the number of microstates is reduced to two (a preset minimum) thereafter, an optimal number of microstates can be selected (Khanna et al., 2014; Poulsen et al., 2018). The number of microstates for each participant were selected using the Krzanowski-Lai criterion applied to the Correlation Sum. This is a method of finding the point past which the Correlation Sum plateaus with respect to the number of microstates (Krzanowski & Lai, 1988). This microstate decomposition was performed for each participant individually. The microstates for each participant were then back fitted to their resting state data, and fitted to their TMS EEG data. After this, the Lempel-Ziv complexity was calculated in a 100 ms sliding window and averaged across epochs and subjects before being compared between resting state and TMS EEG recordings. This short window was considered suitable since good temporal resolution was required for this analysis and previous literature has shown that microstate sequences show scale free dynamics (Van De Ville et al., 2010).

To test the changes in metastability for significance, two 250 ms windows of time were defined, before and after the pulse. Then the epoch averaged measures were averaged in the windows of time, yielding a pre and post pulse metastability value for each participant. Given that the sample size was less than 30 and the results of the Shapiro-Wilk test of normality were inconsistent, normality could not be safely assumed. Thus the difference between the two lists of metastability measures was tested for significance using Wilcoxon's Signed Rank test. This is a non-parametric test suitable for testing dependent samples. The significance threshold was set to 0.05, and p values below this were used to indicate a significant difference. The test was carried out in a one-tailed manner, with the direction being dependent on the effect in question.

2.4 Event Related Potentials

Event Related Potentials (ERPs) are EEG changes that are time locked to sensory, motor or cognitive events (Sur & Sinha, 2009). They are obtained through averaging activity around an event across a large number of trials. The ERPs observed during a specific task can be used to indicate which neural events took place during the task based on what those ERPs were previously associated with. To understand preliminary results showing changes in metastability prior to the TMS pulse, an analysis of event related potential was also performed. Specifically, slow negative waves, which are documented signs of anticipation were tested for using established procedures (Luck & Kappenman, 2011). This analysis consisted of lowpass filtering the data at 2 Hz, epoching around the pulse and then averaging across epochs and participants. 200 milliseconds preceding the TMS pulse were then visually inspected for a slow negative wave.

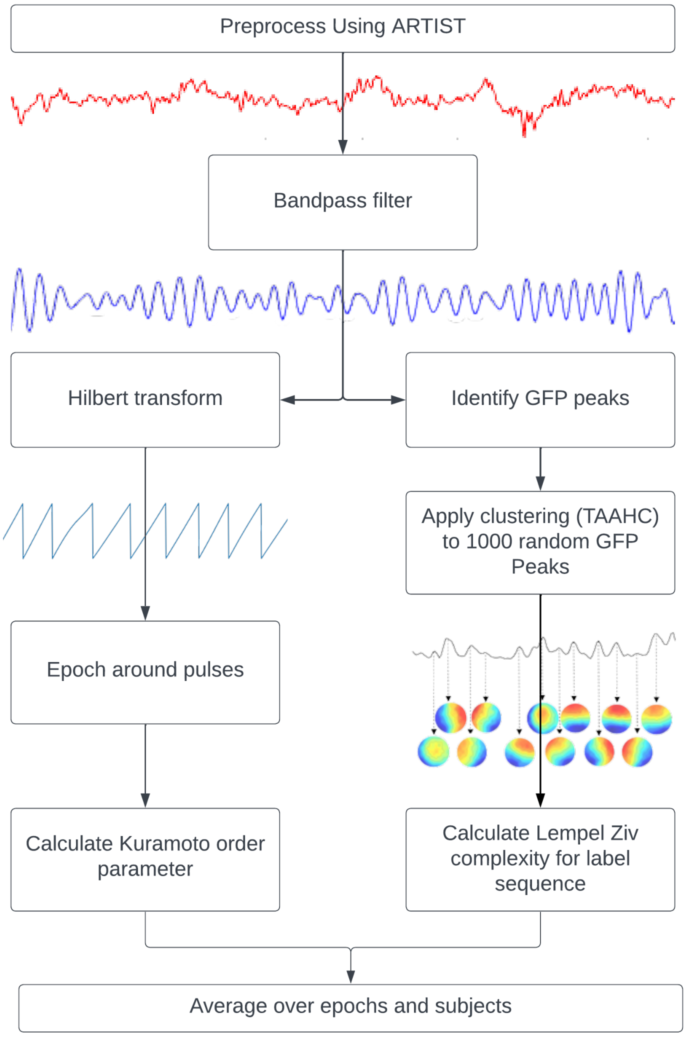
Preprocess Using ARTIST
Bandpass filter
Hilbert transform
Identify GFP peaks
Apply clustering (TAAHC) to 1000 random GFP Peaks
Epoch around pulses
Calculate Kuramoto order parameter
Calculate Lempel Ziv complexity for label sequence
Average over epochs and subjects

Figure 2.3: Pipelines used to calculate the Kuramoto Order Parameter based measure (left) and the Microstate based measure (right)

Chapter 3

Results

3.1 Local Synchronisation Following TMS

The channel grouping algorithm described in the methods section showed a frontal-central cluster and a temporal-occiptial cluster. The groups appear to be clustered around the point of stimulation (rPMC) and probably resulted from the shape of the electrical field induced. The groups created are depicted below (figure 3.1).

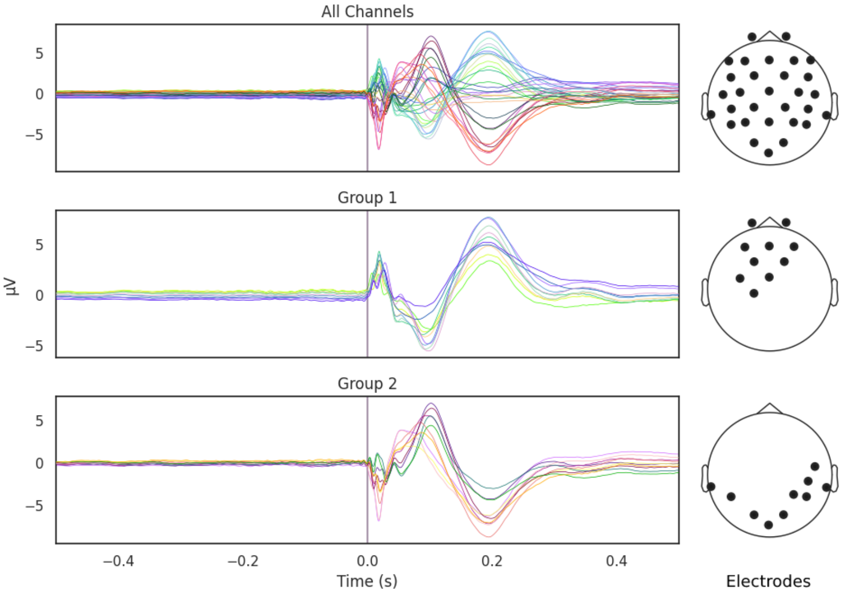

Figure 3.1: Channel groupings derived from the TMS Evoked Potential

3.2 Metastability Fluctuations around the TMS pulse

As stated previously, the metastability analysis based on the Kuramoto Order Parameter was performed for each group to see the effects of local synchrony. Among all 3 groups, spTMS stimulation causes 10 to 40% reduced order parameter standard deviation for a duration dependent on the frequency band. Higher frequencies (alpha, beta, gamma) recover quickly (within 200 ms) while lower frequencies (delta, theta) recover slowly (within 400 ms). In all frequency bands a 10% increase prior to the pulse is seen, while in the alpha and theta bands subsequent recovery 10 to 15% past baseline levels was also observed. At a global level these changes were accompanied by a 10% decrease in coherence, however, for groups 1 and 2, a sharp increase in coherence was observed concomitant with the same changes to metastability. These effects were replicated across all stimulation intensities (120%, 110% and 100% of Resting Motor Threshold). The results for group 1 and 120%RMT TMS (figure 3.2) and the table detailing the significance of each effect (figure 3.3) is given below. The same figures for the other channel groups and stimulation intensities can be found in Appendix A.

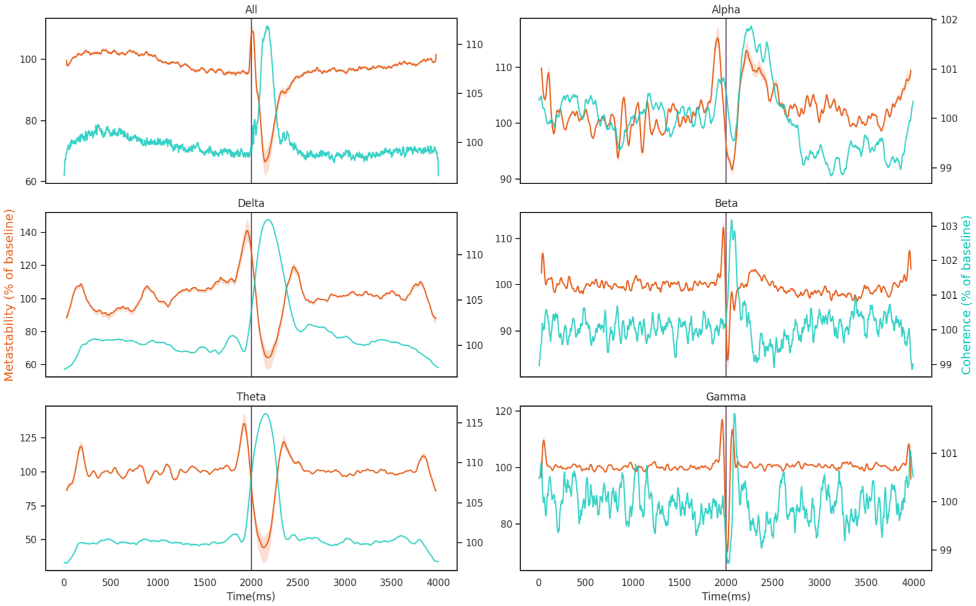

Figure 3.2: Results for the Kuramoto Order Parameter measure of metastability. In each figure the orange line indicates metastability and is scaled on the left y axis, the blue line indicates coherence and is scaled on the right y axis and the TMS pulse is indicated by the vertical purple line. Both metastability and coherence are plotted as a percentage of a baseline value calculated as the mean between 525 and 1525 ms. Results are plotted separately for each frequency band.

	Delta	Theta	Alpha	Beta	Gamma	All
Decrease in KOP after pulse	200 (p<0.001)	210 (p<0.001)	88 (p=0.73)	146 (p=0.06)	202 (p<0.001)	205 (p<0.001)
Increase in KOP before pulse	6 (p<0.001)	2 (p<0.001)	21 (p<0.001)	26 (p<0.001)	13 (p<0.001)	149 (p=0.951)
Increase in KOP after pulse	162 (p=0.985)	55 (p=0.031)	17 (p<0.001)	48 (p=0.016)	82 (p=0.204)	197 (p=0.999)

Figure 3.3: Test statistics and p values for one-tailed Wilcoxon's Signed Rank tests conducted on the Kuramoto Order Parameter measure (KOP).

Microstate global explained variance was worse for the backfitted TMS data than for the resting state data (on which the clustering algorithm was run) but the performance was sufficient for continued analysis (0.65 to 0.60). Microstate analysis shows increased variation in microstate duration, and polarization of transition probabilities in the TMS stimulated data. The Lempel-Ziv complexity increased prior to the TMS pulse (test statistic = 22, p= 0.002) and was reduced following TMS stimulation (test statistic = 138, p value = 0.01) and recovered to baseline levels within 200 ms.

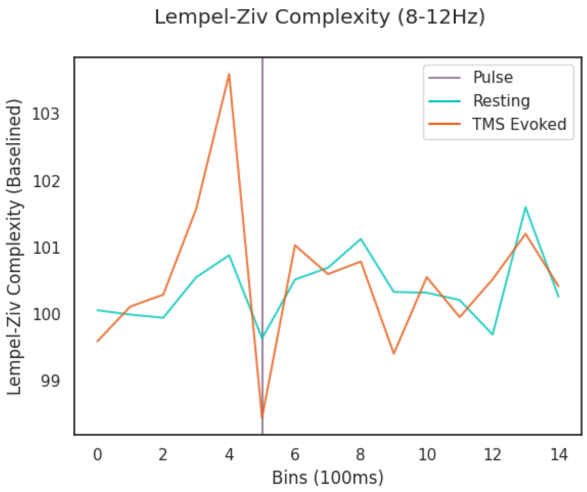

Figure 3.4: Results for the Lempel-Ziv Complexity Measure. The orange line is LZC in the TMS stimulated condition while the blue line is LZC in the resting state. Complexity was calculated in 100 ms bins and averaged across subjects and epochs. The resting state data was subsampled and averaged in the same way. Results are plotted as a percentage of the mean LZC across the first 3 bins.

The transition probabilities for the microstates were initially relatively consistent, with

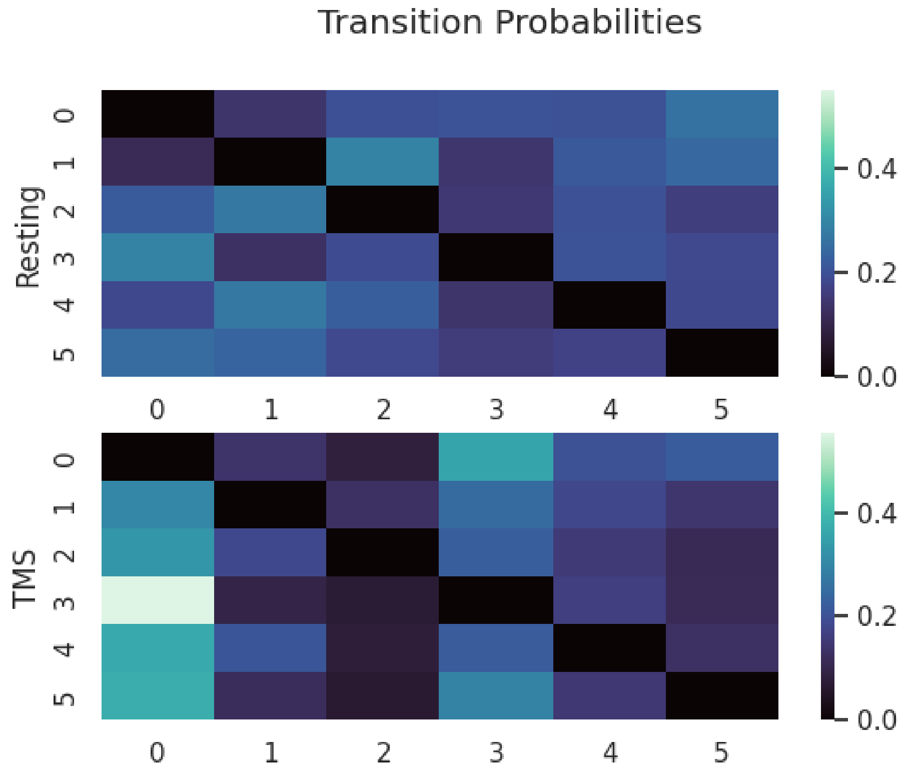

Figure 3.5: Microstate Transition Probabilities. The probability of transitioning from the row microstate to the column microstate is given by the color of the cell as defined by the color map on the right. The bottom heatmap depicts transition probabilities for 1000 ms following TMS, while the top heatmap depicts the same for an equivalent length of resting state data from a given participant.

each microstate have a roughly even chance of transitioning to any other microstate. In the period after the TMS pulse for which metastability is reduced, transition probabilities increase along some columns and decrease along others. Since the columns indicate the probability of another microstate transitioning into a given microstate, this change reflects the repeated consolidation of dynamics to the same pattern(s) of activity.

3.3 Stimulus Preceding Negativity before the TMS pulse

The ERP analysis revealed a Stimulus Preceding Negativity (SPN) with a central distribution prior to the TMS pulse in all 3 stimulation intensities. The reduction in potential observed was between 0.3 and 0.6 microvolts. The small magnitude of this effect could be explained by the aggressive Preprocessing made necessary by TMS artifacts since this significantly attenuates the signal.

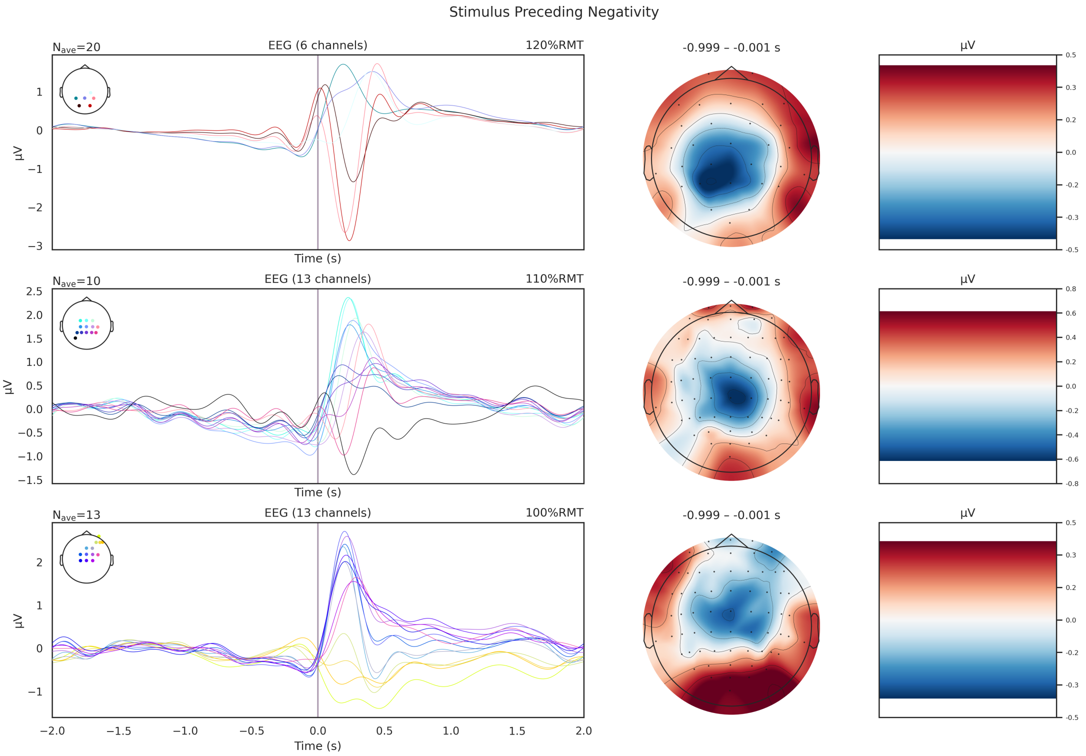

Figure 3.6: Stimulus Preceding Negativity. The voltage timeseries is given for each electrode involved in the SPN (shown in the top left of each voltage timeseries). The spatial distributions are averaged over 1 second prior to the TMS pulse. Results are provided for each stimulation intensity (indicated on the top right of each voltage timeseries.)

Chapter 4

Discussion

4.1 Metastability as a Dynamic Index of Neural Activity

Metastability emerges when a delicate balance between integrative and segregative tendencies exists (Tognoli & Kelso, 2014). It can be understood as a winnerless competition between a set of coordination states (Deco et al., 2017). Since the TMS pulse artificially synchronises certain groups of oscillators, it would follow that the natural balance between them would be disrupted and the dynamics would appear less metastable. Although this effect would be highly localised and brief due to the stimulation alone, the impulse delivered to the stimulated population could propogate through the network and reset their phase dynamics as it does so. This would effectively cluster neural populations based on their delay with the stimulated region, and increase synchronisation within those clusters. The relative simplicity and stability of the resulting coordination state would manifest in reduced metastability. The time taken to return to resting metastable dynamics would then depend on frequency. Since the metastable dynamics in question are those of relative phase, a faster changing phase (frequency) would enable the natural dynamics to reemerge faster. Indeed, this frequency dependency is reflected in the results. The reduced Lempel-Ziv complexity following stimulation corroborates the results of the Kuramoto order parameter based analysis. Furthermore, the polarised transition probabilities between microstates suggests that the brain repeatedly consolidates to the same coupling state under reduced metastability. This aligns with how natural dynamics would periodically start to emerge but then be interrupted as the phase reset propogates thorugh the network with a delay. This set of results supports the idea that metastability can be meaningfully quantified over a 50ms sliding window, and its fluctuations index the coordination dynamics of the network.

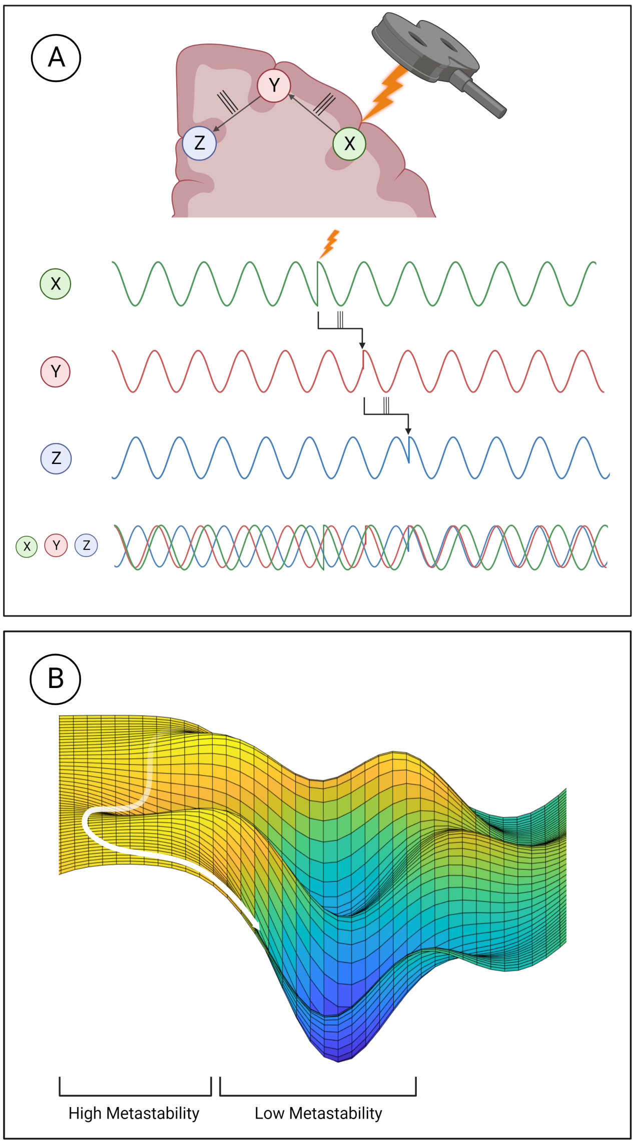

Figure 4.1: Mechanistic explanation of how TMS perturbs metastability. Figure A depicts how the phase locking caused by TMS can propogate through the network, increasing coherence within clusters and decreasing the overall metastability. Figure B shows metastability in terms of an energy landscape. The system has high metastability when it moves in and out of points of stability (wells in the energy landscape) the period of depressed metastability is analogous to a valley, where movement is more constrained.

4.2 Metastability and Anticipation

An interesting and surprising result of this analysis was the increase in metastability observed prior to the pulse in the alpha and theta bands. Having thoroughly ruled out any artifactual sources of this effect, the most likely explanation was anticipation. Since hundreds of pulses were delivered during recording sessions, the distinctive 'click' of the TMS coil was not masked and the interstimulus interval was consistent, it was possible that the subjects came to anticipate the pulse and the increase in metastability reflected that mental state. to address this question, the data were examined for the presence of a Stimulus Preceding Negativity (SPN). The SPN is an ERP associated with the anticipation of an affective or physiologically arousing stimulus such as opposite sex nudes or painful electric shocks (Luck & Kappenman, 2011). In one study, an electric shock was delivered 100 - 300ms after an audio cue, and the SPN was observed in the time after the audio cue (Tanovic & Joormann, 2019). These conditions are highly reminiscent of the TMS coil click being followed by the scalp sensations of TMS. The presence of the SPN in the data and the fact that TMS can produce sensations similar to electrical stimulation, supports the idea that participants were anticipating the stimulus in the same time window as the increase in metastability. The neurophysiological mechanism for this effect and whether the increase in metastability is facilitatory or collateral to anticipation is an area of future investigation.

4.3 Implications and Future Directions

The validation of metastability as a measure of coordination dynamics in short timescales in the primary contribution of this work. Quantified in this way, metastability could be used to index dynamical shifts following any type of intervention to better understand its effects. This is especially useful in the context of non invasive brain stimulation, since the individual variability for a given protocol could be understood and reduced by indexing neural dynamics after stimulation. Furthermore, given how metastability is deranged in a wide range of neurological disorders, stimulation protocols aimed at treating them could show better effects if optimised based on their effect on metastability. It bears advantages over simpler measures such as coherence because of its relevance to pathology and its occurrence at multiple timescales. This property is exceptionally useful since it allows this measure to be applied to both EEG and MRI data, and capture changes occurring over disparate timescales.

Another intriguing implication of this work is that metastability may change in tandem with emotional or cognitive states, as indicated by the increase during the SPN. Studying how metastability changes during cognitive tasks and in response to affective stimuli would shed light on theoretical questions of how mood can affect cognition and how dynamical properties like metastability relate to cognitive function.

Finally, the global desynchronisation and local synchronisation created by TMS presents a unique lens with which to analyse it's effects. Rather than seeing TMS as tool to excite one region, it might instead be seen as a way of linking a set of regions. Building a computational model of this phenomenon and using it to contextualise the effects of various TMS protocols is a direction of future study.

Appendix A

Additional Figures

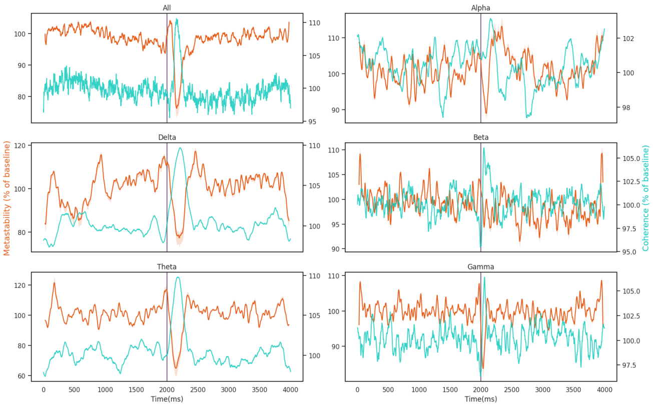

Figure A.1: KOP results for channel group 1 and 100% RMT stimulation

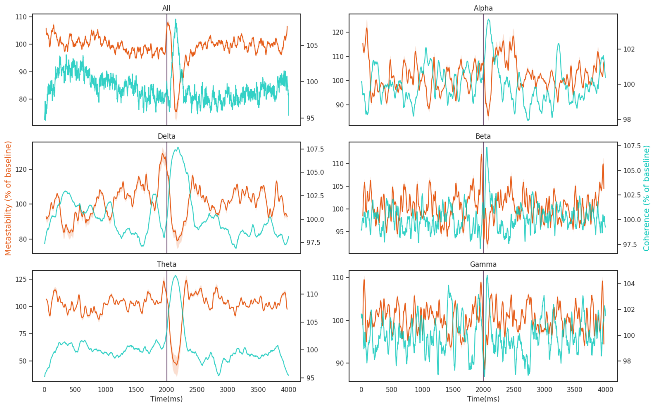

Figure A.2: KOP results for channel group 1 and 110% RMT stimulation

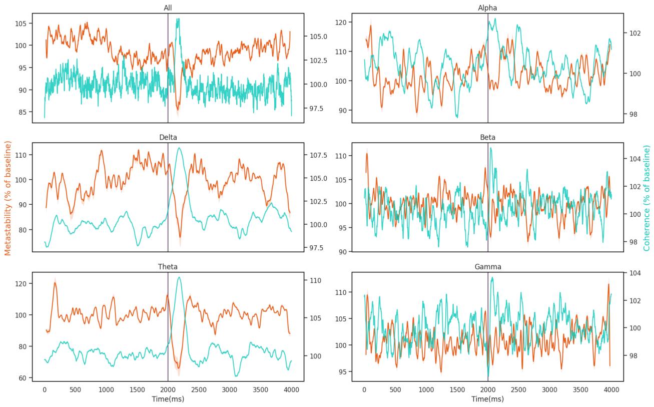

Figure A.3: KOP results for channel group 2 and 100% RMT stimulation

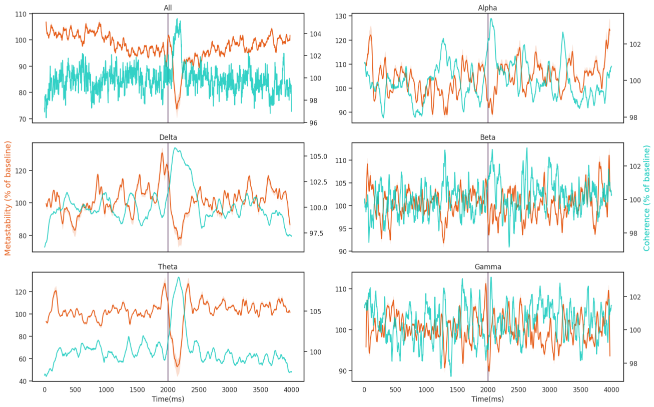

Figure A.4: KOP results for channel group 2 and 110% RMT stimulation

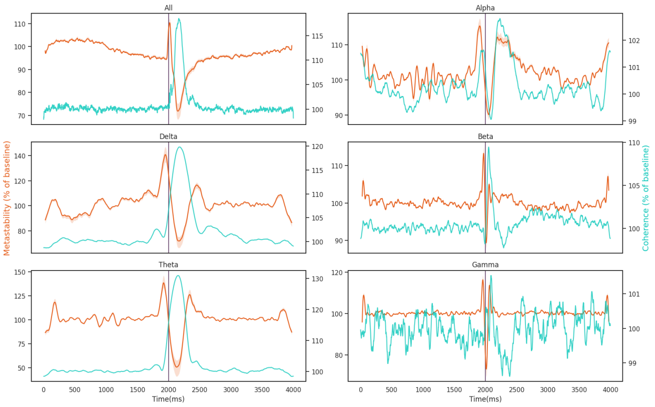

Figure A.5: KOP results for channel group 2 and 120% RMT stimulation

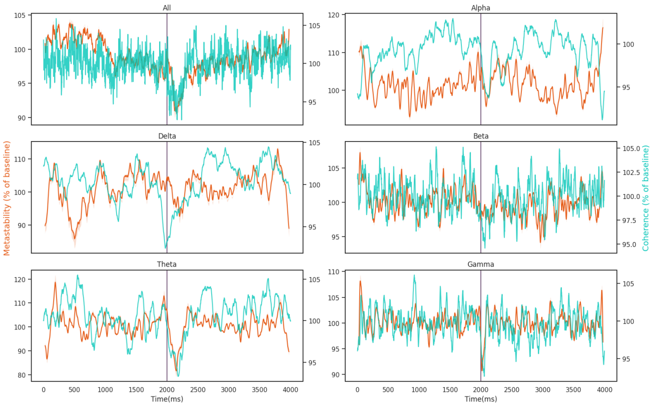

Figure A.6: KOP results for all channels and 100% RMT stimulation

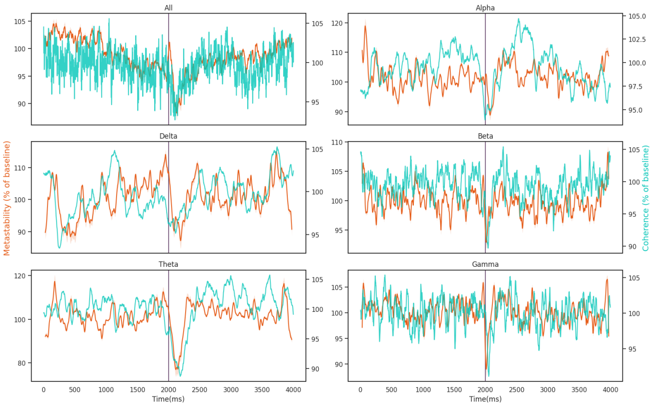

Figure A.7: KOP results for all channels and 110% RMT stimulation

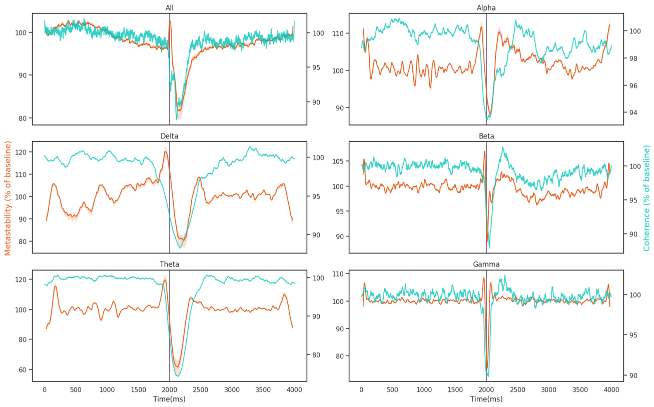

Figure A.8: KOP results for all channels and 120% RMT stimulation

Appendix B

Code Used for Preprocessing, Analysis and Figures

B.1 Python Code used for Analysis and Figures

```python
import numpy as np
import pandas
import mne
import re
import scipy.stats
import scipy.io
import matplotlib.pyplot as plt
import matplotlib.lines as mlines
from pathlib import Path
import seaborn as sns
import antropy
mne.viz.use_browser_backend('qt')

#functions
def coherence_timeseries(phase_timeseries): #voltage_epochxchannelxtime to coherence_epochxtime
    theta_nxt = np.angle(phase_timeseries)
    average_x_t = np.mean(np.cos(theta_nxt), axis = 1)
    average_y_t = np.mean(np.sin(theta_nxt), axis = 1)
    coherence_t = (average_x_t ** 2 + average_y_t ** 2) ** 0.5
    return coherence_t

def phase_coherence(angles_vec): #alternative coherence function
    suma = sum([(np.e ** (1j * i)) for i in angles_vec])
    return abs(suma / len(angles_vec))

def metastability_timeseries(coherence_timeseries, window_size, type): #coherence_epochxtime to metastability_epochxtime
    if type == 'sliding_future':
        indexes = np.arange(coherence_timeseries.shape[1] - window_size)
        meta_t = np.zeros((coherence_timeseries.shape[0], indexes.shape[0]))
        for i in indexes:
            meta_t[:,i] = np.std(coherence_timeseries[:, i:i+window_size], axis=1)
    elif type == 'sliding_past':
        indexes = np.arange(window_size, coherence_timeseries.shape[1])
        meta_t = np.zeros((coherence_timeseries.shape[0], indexes.shape[0] + window_size))
        for i in indexes:
            meta_t[:,i] = np.std(coherence_timeseries[:, i-window_size:i], axis=1)
    elif type == 'sliding_sym':
        interval = int(window_size / 2)
        indexes = np.arange(interval, coherence_timeseries.shape[1] - interval)
        meta_t = np.zeros((coherence_timeseries.shape[0], indexes.shape[0] + interval))
        for i in indexes:
            i = int(i)
            start = int(i - interval)
            end = int(i + interval)
            meta_t[:,i] = np.std(coherence_timeseries[:, start:end], axis=1)
    elif type == 'sliding_pulse_start_sym':
        interval = int(window_size / 2)
        pulse = (coherence_timeseries.shape[1] - 1) / 2
        first_indexes = np.arange(interval, pulse - interval)
        second_indexes = np.arange(pulse + interval, coherence_timeseries.shape[1] - interval)
        indexes = np.concatenate((first_indexes, second_indexes))
        meta_t = np.zeros((coherence_timeseries.shape[0], indexes))
        for i in indexes:
            i = int(i)
            start = int(i - interval)
            end = int(i + interval)
            meta_t[:,i] = np.std(coherence_timeseries[:, start:end], axis=1)
    elif type == 'sliding_pulse_start':
        pulse = (coherence_timeseries.shape[1] - 1) / 2
        indexes = np.arange(window_size, coherence_timeseries.shape[1] - window_size)
        meta_t = np.zeros((coherence_timeseries.shape[0], indexes.shape[0]))
        for i, index in enumerate(indexes):
            if index < pulse:
                meta_t[:,i] = np.std(coherence_timeseries[:, int(index-window_size):int(index)], axis=1)
            else:
                meta_t[:,i] = np.std(coherence_timeseries[:, int(index):int(index+window_size)], axis=1)
    elif type == 'binned':
        pulse = (coherence_timeseries.shape[1] - 1) / 2
        bins = int(coherence_timeseries.shape[1] / window_size)
        meta_t = np.zeros((coherence_timeseries.shape[0], bins))
        for i in range(bins):
            start = int(i * window_size)
            end = int(start + window_size)
            meta_t[:,i] = np.std(coherence_timeseries[:, start:end], axis=1)
    return meta_t

def LZ(x):
    #run length encoding
    where = np.flatnonzero
    x = np.asarray(x)
    n = len(x)
    starts = np.r_[0, where(~np.isclose(x[1:], x[:-1], equal_nan=True)) + 1]
    lengths = np.diff(np.r_[starts, n])
    values = x[starts]
    runs = list(zip(lengths, values))
    #count unique runs
    no_dupes = set(runs)
    uniques = len(no_dupes)
    return uniques

def LZtimeseries(labels_t, window_size, type): #calculates running LZ complexity using provided increment
    timepoints = int(len(labels_t)/window_size)
    LZ_t = np.zeros(timepoints)
    if type == 'incremented':
        for i in range(timepoints):
            LZ_t[i] = antropy.lziv_complexity(labels_t[:window_size + window_size*i], normalize=False)
    if type == 'sliding':
        interval = int(window_size / 2)
        indexes = np.arange(interval, labels_t.shape[0] - interval)
        LZ_t = np.zeros(indexes.shape[0] + interval)
        for i in indexes:
            i = int(i)
            start = int(i - interval)
            end = int(i + interval)
            LZ_t[i] = antropy.lziv_complexity(labels_t[start:end], normalize=False)
    if type == 'binned':
        bins = int(labels_t.shape[0] / window_size)
        LZ_t = np.zeros(bins)
        for i in range(bins):
            LZ_t[i] = LZ(labels_t[i*window_size: i*window_size + window_size])
    return LZ_t

#theme
pal = {"Tiffany␣Blue":"#00C6B8","Persimmon":"#E65C18","Dark␣Slate␣Gray":"#1a5e63","Alice␣Blue":"#f6fafd", "Alice␣Blue␣dark":"#eef6fb", "English␣Violet":
sns.set_theme(context='notebook', style='white')

#path and montage
directory120 = '/home/cbdl/Downloads/Datasets/TMS_EEG/derivatives/2sepoch_nobl_tagged'
pathlist120 = list(Path(directory120).glob('*.fif'))

directory110 = '/home/cbdl/Downloads/Datasets/TMS_EEG_2/derivatives/2sepoch_nobl_tagged/14ms'
pathlist110 = list(Path(directory110).glob(f'*110rmt_eeg.fif'))
```

```
directory100 = '/home/cbdl/Downloads/Datasets/TMS_EEG_2/derivatives/2sepoch_nobl_tagged/14ms'
pathlist100 = list(Path(directory100).glob(f'*100rmt_eeg.fif'))

montage = mne.channels.make_standard_montage('standard_1005')
samples = {'120%RMT': pathlist120, '110%RMT': pathlist110, '100%RMT': pathlist100}

#channel groups
all_sample_groups = {}
for sample, pathlist in samples.items():
    evoked = []
    for path in pathlist:
        raw = mne.io.read_raw_fif(path, preload=True)
        new_names = dict((ch_name, ch_name.rstrip('.').upper().replace('Z', 'z').replace('FP', 'Fp')) for ch_name in raw.ch_names)
        raw.rename_channels(new_names)
        raw.set_montage(montage)
        raw.interpolate_bads()
        events, event_ids = mne.events_from_annotations(raw)
        epochs = mne.Epochs(raw, events, [3], baseline=(None,-1), tmin=-2, tmax=2) #'S  3'
        evoked.append(epochs.average())
    evoked_gr = mne.grand_average(evoked)
    evoked_cxt = evoked_gr.get_data()
    evoked_std_t = np.std(evoked_cxt, axis = 0)
    peak_std = np.where(evoked_std_t == np.max(evoked_std_t))
    evoked_peak_c = evoked_cxt[:,peak_std]
    mdiff_c = evoked_peak_c.mean() - evoked_peak_c
    channel_fraction = 10/mdiff_c.shape[0]
    group1 = np.where(mdiff_c < np.quantile(mdiff_c, channel_fraction))[0]
    group2 = np.where(mdiff_c > np.quantile(mdiff_c, 1 - channel_fraction))[0]
    all_sample_groups[f'{sample}_group1'] = group1
    all_sample_groups[f'{sample}_group2'] = group2
    all_sample_groups[f'{sample}_all'] = None
    if sum(group1) > sum(group2):
        all_sample_groups[f'{sample}_group1'] = group2
        all_sample_groups[f'{sample}_group2'] = group1
    evoked_gr.plot(gfp = True, spatial_colors = True)
    evoked_gr.plot(gfp = True, spatial_colors = True, picks=group1)
    evoked_gr.plot(gfp = True, spatial_colors = True, picks=group2)

#metastability and coherence calculation
frequencies = {'all':None, 'delta':[1,3], 'theta':[4,7], 'alpha':[8,12], 'beta':[15,30], 'gamma':[35,42]}
window = 50
epoch_avg_coherence = {}
epoch_avg_meta = {}
gr_coherences = {}
gr_meta = {}
meta_std = {}
for sample, pathlist in samples.items():
    sample_groups = {name:group for name, group in all_sample_groups.items() if sample in name}
    for index, path in enumerate(pathlist):
        raw = mne.io.read_raw_fif(path, preload=True)
        new_names = dict((ch_name, ch_name.rstrip('.').upper().replace('Z', 'z').replace('FP', 'Fp')) for ch_name in raw.ch_names)
        raw.rename_channels(new_names)
        raw.set_montage(montage)
        raw.interpolate_bads()
        events, event_ids = mne.events_from_annotations(raw)
        print(f'retreived␣participant␣{index}')
        for freq, value in frequencies.items():
            if freq != 'all':
                raw_filt = raw.copy().filter(value[0], value[1])#, l_trans_bandwidth=value[1]*0.25, h_trans_bandwidth=value[1]*0.25
            else:
                raw_filt = raw.copy()
            raw_filt.apply_hilbert()
            epochs = mne.Epochs(raw_filt, events, [3], baseline=(None,-1), tmin=-2, tmax=2, preload=True) #10001
            print(f'{freq}␣data␣transformed␣and␣epoched')
            for group in ['group1', 'group2', 'all']:
                voltage_excxt = epochs.get_data(picks=sample_groups[f'{sample}_{group}'])
                coherence_ext = coherence_timeseries(voltage_excxt)
                coherence_avg_t = coherence_ext.mean(axis=0)
                meta_ext = metastability_timeseries(coherence_ext, window, 'sliding_past')
                meta_avg_t = meta_ext.mean(axis=0)
                epoch_avg_coherence[f'{sample}_{index}_{freq}_{group}'] = coherence_avg_t
                epoch_avg_meta[f'{sample}_{index}_{freq}_{group}'] = meta_avg_t
                print(f'grouped␣and␣epoched␣data␣of␣shape:␣{voltage_excxt.shape}')
                print(f'epoch-wise␣coherence␣of␣shape:␣{coherence_ext.shape}␣epoch␣averaged␣coherence␣of␣shape:␣{coherence_avg_t.shape}')
                print(f'epoch-wise␣metastability␣of␣shape:␣{meta_ext.shape}␣epoch␣averaged␣metastability␣of␣shape:␣{meta_avg_t.shape}')

    #subject averaging
    meta_arr = np.zeros([len(pathlist), epoch_avg_meta[f'{sample}_0_all_all'][epoch_avg_meta[f'{sample}_0_all_all'] != 0].shape[0]])
    for frequency in frequencies:
        for group in ['group1', 'group2', 'all']:
            keys = [key for key in epoch_avg_meta.keys() if key.split('_')[0] == sample and key.split('_')[2] == frequency and key.split('_')[3] == grou
            gr_avg_meta = np.zeros(epoch_avg_meta[f'{sample}_0_alpha_all'].shape[0])
            gr_avg_coherence = np.zeros(epoch_avg_coherence[f'{sample}_0_alpha_all'].shape[0])
            print(len(keys), f'{frequency}_{group}')
            for i, key in enumerate(keys):
                gr_avg_meta += epoch_avg_meta[key]
                gr_avg_coherence += epoch_avg_coherence[key]
                meta_arr[i,:] = epoch_avg_meta[key][epoch_avg_meta[key] != 0]
            gr_avg_meta /= len(keys)
            gr_avg_coherence /= len(keys)
            gr_coherences[f'{sample}_{frequency}_{group}'] = gr_avg_coherence
            gr_meta[f'{sample}_{frequency}_{group}'] = gr_avg_meta
            meta_std[f'{sample}_{frequency}_{group}'] = np.std(meta_arr, axis=0)

#Produces KOP measure figure
pulse = 2000
baseline = [525,1525]
window=50
for group in ['all', 'group1', 'group2']:
    for stim in ['120%RMT', '110%RMT', '100%RMT']:
        path = f'/home/cbdl/Downloads/Datasets/figures/RM1_new_final/{group}_{stim}.png'
        fig, axs = plt.subplots(nrows=3, ncols=2, sharex='col', figsize=(16,10), tight_layout=True, gridspec_kw={'width_ratios': [1, 1]})
        axs[2,0].set_xlabel('Time(ms)')
        axs[2,1].set_xlabel('Time(ms)')
        axs[1,0].set_ylabel(f'Metastability␣(%␣of␣baseline)', size=14, color=pal['Persimmon'])
        keys = [key for key in gr_meta.keys() if key.split('_')[0] == stim and key.split('_')[2] == group]
        #axs[i,1].plot(0.05,0.01,'x',c=pal["English Violet"])
        for i, key in enumerate(keys):
            frequency = key.split('_')[1]
            meta_std = np.std(gr_meta[key], axis=0)
            baseline_meta = gr_meta[key][baseline[0]:baseline[1]].mean()
            baselined_meta = (gr_meta[key][gr_meta[key] != 0] / baseline_meta) * 100
            baseline_coh = gr_coherences[key][baseline[0]:baseline[1]].mean()
            baselined_coh = (gr_coherences[key][gr_coherences[key] != 0] / baseline_coh) * 100
            meta_x = np.arange(int(window/2), gr_coherences[key].shape[0] - int(window/2))
            meta_plus = gr_meta[key][gr_meta[key] != 0] + meta_std
            meta_plus_baseline = meta_plus[baseline[0]:baseline[1]].mean()
            meta_plus_baselined = (meta_plus / meta_plus_baseline) * 100
            meta_minus = gr_meta[key][gr_meta[key] != 0] - meta_std
            meta_minus_baseline = meta_minus[baseline[0]:baseline[1]].mean()
            meta_minus_baselined = (meta_minus / meta_minus_baseline) * 100
            if i <= 2:
                axs[i,0].set_title(frequency.capitalize())
                axs[i,0].locator_params('y',nbins=5)
                axs[i,0].axvline(pulse, label='pulse', color=pal['English␣Violet'], alpha=0.8)
                axs[i,0].plot(meta_x, baselined_meta, color=pal['Persimmon'])
                axs[i,0].fill_between(meta_x, meta_minus_baselined, meta_plus_baselined, facecolor=pal['Persimmon'], alpha=0.2)
                twin = axs[i,0].twinx()
                twin.locator_params('y',nbins=5)
                twin.plot(baselined_coh, color=pal['Tiffany␣Blue'], alpha=0.8)
            else:
                axs[i-3,1].set_title(frequency.capitalize())
                axs[i-3,1].locator_params('y',nbins=5)
                axs[i-3,1].axvline(pulse, label='pulse', color=pal['English␣Violet'], alpha=0.8)
                axs[i-3,1].plot(meta_x, baselined_meta, color=pal['Persimmon'])
                axs[i-3,1].fill_between(meta_x, meta_minus_baselined, meta_plus_baselined, facecolor=pal['Persimmon'], alpha=0.2)
                twin = axs[i-3,1].twinx()
                twin.locator_params('y',nbins=5)
                twin.plot(baselined_coh, color=pal['Tiffany␣Blue'], alpha=0.8)
            if i == 4:
                twin.set_ylabel(f'Coherence␣(%␣of␣baseline)', size=14, color=pal['Tiffany␣Blue'])
        fig.savefig(path, bbox_inches='tight')

##sig testing
#test windows of time for significant difference
samples = ['120%RMT', '110%RMT', '100%RMT']
normality = {}
paired_t = {}
for sample in samples:
    for frequency in frequencies:
        for group in ['all', 'group1', 'group2']:
            keys = [key for key in epoch_avg_meta if key.split('_')[0] == sample and key.split('_')[2] == frequency and key.split('_')[3] == group]
            pre_250 = np.zeros((len(keys), 250))
            post_250 = np.zeros((len(keys), 250))
            for i, key in enumerate(keys):
                pre_250[i,:] = epoch_avg_meta[key][1500:1750]
                post_250[i,:] = epoch_avg_meta[key][1750:2000]
            pre_means = pre_250.mean(axis=1)
            post_means = post_250.mean(axis=1)
            print(f'{pre_250.shape},␣{post_250.shape},␣{pre_means.shape},␣{post_means.shape}')
            normal = scipy.stats.shapiro(pre_means)[1] > 0.05 and scipy.stats.shapiro(post_means)[1] > 0.05
            result = scipy.stats.wilcoxon(pre_means, post_means, axis=0, alternative='less')
            paired_t[f'{sample}_{frequency}_{group}'] = {'tstat': result.statistic, 'pvalue': result.pvalue, 'normality': normal}

#significant readout based on given alpha
alpha = 0.05
for key in paired_t:
    if paired_t[key]['pvalue'] < alpha:
        p = paired_t[key]['pvalue']
        print(f'{key}␣is␣significant␣(p_value␣=␣{p})')
    else:
        p = paired_t[key]['pvalue']
        print(f'{key}␣is␣not␣significant␣(p_value␣=␣{p})')

##Microstate analysis
#pathlist
tep_directory = '/home/cbdl/Downloads/Datasets/TMS_EEG/derivatives/variables/tep_microstates/alpha/'
tep_directory = '/home/cbdl/Downloads/Datasets/TMS_EEG_2/derivatives/tep_microstates/recleaned_concatenated/100rmt'
tep_pathlist = list(Path(tep_directory).glob('*.mat'))
tep_pathlist = list(Path(tep_directory).glob('*110_rm1_whole_ms.mat'))

rest_directory = '/home/cbdl/Downloads/Datasets/TMS_EEG/derivatives/variables/resting_microstates/alpha'
rest_directory = '/home/cbdl/Downloads/Datasets/TMS_EEG_2/derivatives/resting_microstates/1khz_resample'
rest_pathlist = list(Path(rest_directory).glob('*.mat'))

#match names
tep_names = [str(path).split('/')[-1].split('_')[0] for path in tep_pathlist]
rest_names = [str(path).split('/')[-1].split('_')[0] for path in rest_pathlist]
rest_final = []
tep_final = []
for i, name in enumerate(rest_names):
    if name in tep_names and name not in ['768','779']:
        rest_final.append(rest_pathlist[i])
        tep_final.append(tep_pathlist[i])
tep_final.sort()
rest_final.sort()

#load matrices in dict
tep_microstates = {}
resting_microstates = {}
for i, path in enumerate(tep_final):
    tep_microstates[i] = scipy.io.loadmat(path, simplify_cells=True)
    resting_microstates[i] = scipy.io.loadmat(rest_final[i], simplify_cells=True)

#transition heatmap
path = '/home/cbdl/Downloads/Datasets/TMS_EEG_2/derivatives/figures/ms_heatmaps/100_restvswhole'
path = '/home/cbdl/Downloads/Datasets/TMS_EEG/derivatives/figures/ms_heatmaps/alpha/taahc_pinv'
path = '/home/cbdl/Downloads/Datasets/figures/RM1_new_final/ms_heatmaps'
for i in tep_microstates:
    fig, axs = plt.subplots(2)
    stat1 = resting_microstates[i]['stats']['TP']
    stat2 = tep_microstates[i]['stats']['avgs']['TP']
    sns.heatmap(stat1, ax=axs[0], vmin=0, vmax=stat2.max(), cmap='mako')
    sns.heatmap(stat2, ax=axs[1], vmin=0, vmax=stat2.max(), cmap='mako')
    fig.suptitle('Transition␣Probabilities')
    axs[0].set_ylabel(f'Resting')
    axs[1].set_ylabel(f'TMS')
    fig.savefig(path + f'/{i}_heatmap.png')

#LZ difference
LZvs = np.zeros(len(tep_microstates))
for i in tep_microstates:
    rstart = np.random.randint(0,resting_microstates[i]['fit']['labels'].shape[0] - 1000)
    rest = resting_microstates[i]['fit']['labels'][rstart:rstart + 1000]
    tep = tep_microstates[i]['fit']['labels'][:,3000:4000]
    LZrest = LZ(rest)
    LZtep = np.zeros(tep.shape[0])
    for ii, epoch in enumerate(tep):
        LZtep[ii] = LZ(epoch)
    LZtep = np.mean(LZtep)
    LZvs[i] = LZtep - LZrest

#LZ trace plot
```

```
trace_count = 100
for i in tep_microstates:
    fig = plt.figure()
    tep_labels_ext = tep_microstates[i]['fit']['labels']
    for ii in range(trace_count):
        r_epoch = np.random.choice(tep_labels_ext.shape[0])
        tep_LZ_t = LZtimeseries(tep_labels_ext[r_epoch,:], 100, type='incremented')
        tep_slopes = np.diff(tep_LZ_t) / np.diff(np.arange(tep_LZ_t.shape[0]))
        plt.plot(tep_LZ_t, alpha=0.2, color=pal["Persimmon"])
        r_start = np.random.choice(resting_microstates[i]['fit']['labels'].shape[0])
        rest_labels_t = resting_microstates[i]['fit']['labels'][r_start:r_start+tep_labels_ext.shape[1]]
        rest_LZ_t = LZtimeseries(rest_labels_t, 100, type='incremented')
        rest_slopes = np.diff(rest_LZ_t) / np.diff(np.arange(rest_LZ_t.shape[0]))
        plt.plot(rest_LZ_t, alpha=0.2, color=pal['Tiffany␣Blue'])
    fig.savefig(f'/home/cbdl/Downloads/Datasets/figures/RM1_LZtimeseries/taahc_pinv/{i}.png')

bin_size = 100
gr_tep_LZ_t = np.zeros((len(tep_microstates), int(tep_microstates[0]['fit']['labels'][:,1500:3000].shape[1]/bin_size)))
gr_rest_LZ_t = np.zeros_like(gr_tep_LZ_t)
for i in tep_microstates:
    if i not in [0,13,15]:
        tep_labels_ext = tep_microstates[i]['fit']['labels'][:,1500:3000]
        rest_labels_t = resting_microstates[i]['fit']['labels']
        tep_LZ_t = np.zeros((tep_labels_ext.shape[0], int(tep_labels_ext.shape[1]/bin_size)))
        rest_LZ_t = np.zeros_like(tep_LZ_t)
        for ii in range(tep_labels_ext.shape[0]):
            tep_LZ_t[ii,:] = LZtimeseries(tep_labels_ext[ii,:], bin_size, type='binned')
            r_start = np.random.choice((resting_microstates[i]['fit']['labels'].shape[0]-tep_labels_ext.shape[1]))
            r_rest = rest_labels_t[r_start:r_start+tep_labels_ext.shape[1]]
            rest_LZ_t[ii,:] = LZtimeseries(r_rest, bin_size, type='binned')
        gr_tep_LZ_t[i] = tep_LZ_t.mean(axis=0)
        gr_rest_LZ_t[i] = rest_LZ_t.mean(axis=0)
gr_tep_baselined = (gr_tep_LZ_t.mean(axis=0) / np.mean(gr_tep_LZ_t.mean(axis=0)[0:3])) * 100
gr_rest_baselined = (gr_rest_LZ_t.mean(axis=0) / np.mean(gr_rest_LZ_t.mean(axis=0)[0:3])) * 100
fig, axs = plt.subplots()
fig.suptitle('Lempel-Ziv␣Complexity␣(8-12Hz)')
pulse = axs.axvline(5, label='Pulse', alpha=0.6, color=pal['English␣Violet'])
resting = axs.plot(gr_rest_baselined, alpha=0.8, color=pal['Tiffany␣Blue'], label='Resting')
tep = axs.plot(gr_tep_baselined, alpha=0.8, color=pal["Persimmon"], label='TMS␣evoked')
axs.set_xlabel('Bins␣(100ms)')
axs.set_ylabel('Lempel-Ziv␣Complexity␣(Baselined)')
blue_line = mlines.Line2D([],[],color=pal['Tiffany␣Blue'])
orange_line = mlines.Line2D([],[],color=pal['Persimmon'])
axs.legend([pulse, blue_line, orange_line], ['Pulse','Resting','TMS␣Evoked'])
fig.savefig(f'/home/cbdl/Downloads/Datasets/figures/RM1_new_final/LZ_gr.png')

#100 random resting vs after comparisons
LZ_pvals = np.zeros(100)
randrange = np.arange(0,62790 - 1000)
r_starts = np.random.choice(randrange, 100, replace=False)
for i in range(100):
    LZrest = np.zeros(len(tep_microstates))
    LZtep = np.zeros(len(tep_microstates))
    r_start = r_starts[i]
    for ii in range(len(tep_microstates)):
        rest = resting_microstates[ii]['fit']['labels'][r_start:r_start + 1000]
        tep = tep_microstates[ii]['fit']['labels'][:,3000:4000]
        LZrest[ii] = LZ(rest)
        LZepochs = np.zeros(tep.shape[0])
        for iii in range(tep.shape[0]):
            LZepochs[iii] = LZ(tep[iii])
        LZtep[ii] = np.mean(LZepochs)
    _, LZ_pvals[i] = scipy.stats.wilcoxon(LZrest, LZtep, alternative='greater')

sig_fraction = np.mean([pval < 0.05 for pval in LZ_pvals])
mean_pval = np.mean(LZ_pvals)

#comparing before vs after
LZbefore = np.zeros(len(tep_microstates))
LZafter = np.zeros(len(tep_microstates))
for i in range(len(tep_microstates)):
    before = tep_microstates[i]['fit']['labels'][:,1900:2000]
    after = tep_microstates[i]['fit']['labels'][:,2000:2100]
    LZbefore_epoch = np.zeros(before.shape[0])
    LZafter_epoch = np.zeros(after.shape[0])
    for ii in range(before.shape[0]):
        LZbefore_epoch[ii] = LZ(before[ii,:])
        LZafter_epoch[ii] = LZ(after[ii,:])
    LZbefore[i] = LZbefore_epoch.mean()
    LZafter[i] = LZafter_epoch.mean()
stat, pval = scipy.stats.wilcoxon(LZbefore, LZafter, alternative='greater')

##Data concatenation
#create list of participants in 14ms condition
directory = '/home/cbdl/Downloads/Datasets/TMS_EEG_2/derivatives/recleaned_tagged/14ms'
pathlist_14ms = list(Path(directory).glob('*110rmt_eeg.fif'))
names = [re.search('CON\d*',str(path)).group() for path in pathlist]
raw_dict = dict(zip(names,pathlist_14ms))

#find other recordings of those participants
conditions = ['4ms', '9ms']
matched = []
for condition in conditions:
    directory = f'/home/cbdl/Downloads/Datasets/TMS_EEG_2/derivatives/recleaned_tagged/{condition}'
    pathlist_other = list(Path(directory).glob('*110rmt_eeg.fif'))
    matched += [str(path) for path in pathlist_other if re.search('CON\d*', str(path)).group() in names]

for name, raw in raw_dict.items():
    matches = [path for path in matched if name in path]
    raw_dict[name] = [raw] + matches

#concatenate each participant's recordings
for name, paths in raw_dict.items():
    raw_list = [mne.io.read_raw_fif(path, preload=True) for path in paths]
    if len(raw_list)>1:
        print(f'concatenating␣the␣following␣{len(paths)}␣raw␣files')
        for path in paths:
            print(path)
        concatenated_raw = mne.concatenate_raws(raw_list)
        concatenated_raw.save(f'/home/cbdl/Downloads/Datasets/TMS_EEG_2/derivatives/concatenated/{name}_110rmt_eeg.fif')
    else:
        raw_list[0].save(f'/home/cbdl/Downloads/Datasets/TMS_EEG_2/derivatives/concatenated/{name}_110rmt_eeg.fif')

##ERP analysis
directory = '/home/cbdl/Downloads/Datasets/TMS_EEG_2/derivatives/concatenated/100rmt'
directory = '/home/cbdl/Downloads/Datasets/TMS_EEG/derivatives/recleaned_tagged'
pathlist = list(Path(directory).glob('*.fif'))
montage = mne.channels.make_standard_montage('standard_1020')
electrodes = ['Fp1','Fp2','F3','F4','C3','C4']
electrodes = ['Fp1', 'Fpz', 'Fp2', 'AF7', 'AF3', 'AFz', 'AF4', 'Fz', 'F2', 'F4', 'F8', 'F6']
electrodes = ['C3', 'C4']

evoked_gr = {}
channels = {}
for sample, pathlist in samples.items():
    evoked = []
    for path in pathlist:
        raw = mne.io.read_raw_fif(path, preload=True)
        new_names = dict((ch_name, ch_name.rstrip('.').upper().replace('Z', 'z').replace('FP', 'Fp')) for ch_name in raw.ch_names)
        raw.rename_channels(new_names)
        raw.set_montage(montage)
        raw.interpolate_bads()
        raw.filter(None,2)
        events, event_ids = mne.events_from_annotations(raw)
        epochs = mne.Epochs(raw, events, [3], baseline=(None,-1), tmin=-2, tmax=2, reject_by_annotation=True, picks=None) #'S  3'
        evoked.append(epochs.average())
    evoked_gr[sample] = mne.grand_average(evoked)
    evoked_cxt = evoked_gr[sample].get_data()
    evoked_std_t = np.std(evoked_cxt, axis = 0)
    peak_std = np.where(evoked_std_t == np.max(evoked_std_t[:1800]))
    evoked_peak_c = evoked_cxt[:,peak_std]
    mdiff_c = evoked_peak_c.mean() - evoked_peak_c
    group1 = np.where(mdiff_c < np.quantile(mdiff_c, 0.2))[0]
    group2 = np.where(mdiff_c > np.quantile(mdiff_c, 0.8))[0]
    channels[sample] = group2


#Produces the SPN figure
fig, axs = plt.subplots(nrows=3, ncols=3, sharex='col', figsize=(16,10), gridspec_kw={'width_ratios': [3, 1, 1]})
fig.suptitle('Stimulus␣Preceding␣Negativity', y=1.02)
for i, sample in enumerate(samples):
    evoked = evoked_gr[sample]
    chans = channels[sample]
    evoked.plot(picks=chans, spatial_colors=True, axes=axs[i,0], show=False)
    evoked.plot_topomap(times=-0.5, average=1, axes=[axs[i,1], axs[i,2]], show=False)
    axs[i,0].axvline(0, alpha=0.5, color=pal['English␣Violet'])
    axs[i,0].set_title(sample, loc='right')
fig.savefig(f'/home/cbdl/Downloads/Datasets/figures/RM1_new_final/SPN_3x2.png', bbox_inches='tight', dpi=300)


#Produces the channel grouping figure
fig, axs = plt.subplots(nrows=3, ncols=2, sharex='col', tight_layout=True, gridspec_kw={'width_ratios': [2, 1]})
all_sample_groups = {}
pathlist = samples['120%RMT']
evoked = []
for path in pathlist:
    raw = mne.io.read_raw_fif(path, preload=True)
    new_names = dict((ch_name, ch_name.rstrip('.').upper().replace('Z', 'z').replace('FP', 'Fp')) for ch_name in raw.ch_names)
    raw.rename_channels(new_names)
    raw.set_montage(montage)
    raw.interpolate_bads()
    events, event_ids = mne.events_from_annotations(raw)
    epochs = mne.Epochs(raw, events, [3], baseline=(None,-1), tmin=-2, tmax=2, preload=True) #'S  3'
    epochs.crop(-0.5,0.5)
    evoked.append(epochs.average())
evoked_gr = mne.grand_average(evoked)
evoked_cxt = evoked_gr.get_data()
evoked_std_t = np.std(evoked_cxt, axis = 0)
peak_std = np.where(evoked_std_t == np.max(evoked_std_t))
evoked_peak_c = evoked_cxt[:,peak_std]
mdiff_c = evoked_peak_c.mean() - evoked_peak_c
channel_fraction = 10/mdiff_c.shape[0]
group1 = np.where(mdiff_c < np.quantile(mdiff_c, channel_fraction))[0]
group2 = np.where(mdiff_c > np.quantile(mdiff_c, 1 - channel_fraction))[0]
all_sample_groups[f'{sample}_group1'] = group1
all_sample_groups[f'{sample}_group2'] = group2
all_sample_groups[f'{sample}_all'] = None
if sum(group1) > sum(group2):
    all_sample_groups[f'{sample}_group1'] = group2
    all_sample_groups[f'{sample}_group2'] = group1
montage = mne.channels.make_standard_montage('standard_1020')
raw = mne.io.read_raw_brainvision('TMS_EEG/derivatives/ARTIST/clean/clean_sub-718_task-tmseeg1_eeg.vhdr')
new_names = dict((ch_name, ch_name.rstrip('.').upper().replace('Z', 'z').replace('FP', 'Fp')) for ch_name in raw.ch_names)
raw.rename_channels(new_names)
raw.set_montage(montage)
evoked_gr.plot(axes=axs[0,0], gfp = False, spatial_colors = True, show=False)
raw.plot_sensors(axes=axs[0,1], title='', ch_type='eeg', show=False)
evoked_gr.plot(axes=axs[1,0], gfp = False, spatial_colors = True, picks=group1, show=False)
picks1=raw.copy().pick(all_sample_groups['120%RMT_group1'])
picks1.plot_sensors(axes=axs[1,1], title='', ch_type='eeg', show=False)
evoked_gr.plot(axes=axs[2,0], gfp = False, spatial_colors = True, picks=group2, show=False)
picks2=raw.copy().pick(all_sample_groups['120%RMT_group2'])
picks2.plot_sensors(axes=axs[2,1], title='', ch_type='eeg', show=False)
axs[0,0].axvline(0, alpha=0.5, color=pal['English␣Violet'])
axs[1,0].axvline(0, alpha=0.5, color=pal['English␣Violet'])
axs[2,0].axvline(0, alpha=0.5, color=pal['English␣Violet'])
fig.show()
```

B.2 MATLAB Code used for Preprocessing and Analysis

```
cfg.EventCode = 'S␣␣3' %'R128'
cfg.TrialStart = -2000
cfg.TrialEnd = 2000
cfg.Baseline = ['none']
cfg.TMSEEGrootFolder = '/home/cbdl/Downloads/Datasets/TMS_EEG/derivatives/ARTIST'

paths = dir('/home/cbdl/Downloads/Datasets/TMS_EEG/sub*/eeg/*tmseeg1_eeg.vhdr')

%ARTIST
for i = 1:length(paths)
    EEG = pop_loadbv(paths(i).folder, paths(i).name);
    EEG = pop_select(EEG, nochannel = {'EMG␣Right','EMG␣Left','ECG','VEOG','HEOG'})
    EEG = pop_chanedit(EEG, 'lookup', 'home/cbdl/MATLAB␣Add-Ons/Collections/EEGLAB/functions/supportfiles/Standard-10-5-Cap385.sfp');
    cleanedEEG = ARTISTMain(EEG,cfg);
    com = pop_writebva(cleanedEEG, strcat('/home/cbdl/Downloads/Datasets/TMS_EEG_2/derivatives/ARTIST/clean/', paths(i).name));
end

%ASR + MARA
eeglab;
```

```
for i = 1:length(paths)
    %load
    EEG = pop_loadbv(paths(i).folder, paths(i).name);
    %EEG = pop_select(EEG, nochannel = {'EMG Right','EMG Left','ECG','VEOG','HEOG'})
    EEG = pop_chanedit(EEG, 'lookup', 'home/cbdl/MATLAB␣Add-Ons/Collections/EEGLAB/functions/supportfiles/Standard-10-5-Cap385.sfp');
    %resample
    EEG = pop_resample(EEG, 1000)
    %bandpass filter, electrical noise removal
    EEG_f = pop_eegfiltnew(EEG,1,0)
    EEG_f = pop_eegfiltnew(EEG_f,0,100)
    EEG_f = pop_cleanline(EEG_f, 'SignalType', 'Channels', 'LineFrequencies', '[60]')
    %ASR
    EEG_asr = clean_artifacts(EEG_f, 'Highpass', 'off', 'LineNoiseCritereion', 'off')
    %EEG_asr = clean_artifacts(EEG_f, 'BurstRejection', 'on', 'Highpass', 'off', 'LineNoiseCritereion', 'off')
    %ICA
    [EEG_asr.icaweights, EEG_asr.icasphere] = runica(EEG_asr.data)
    [ALLEEG, EEG_asr, CURRENTSET] = eeg_store(ALLEEG, EEG_asr)
    [~,EEG_ica,~] = processMARA(ALLEEG, EEG_asr, CURRENTSET, [0,0,0,0,0])
    EEG_ica = pop_subcomp(EEG_ica, [], 0, 0)
    %interpolate, rereference
    EEG_int = eeg_interp(EEG_ica, EEG.chanlocs)
    EEG_clean = pop_reref(EEG_int, [])
    %export
    pop_writebva(EEG_clean, strcat('/home/cbdl/Downloads/Datasets/TMS_EEG/derivatives/clean_rest_1k_resample/', paths(i).name));
end

%load data (rest)
eeglab;
for i = 1:length(paths)
    EEG = pop_loadbv(paths(i).folder, paths(i).name);
    EEG = pop_chanedit(EEG, 'lookup', 'home/cbdl/MATLAB␣Add-Ons/Collections/EEGLAB/functions/supportfiles/Standard-10-5-Cap385.sfp');
    %EEG = pop_select(EEG, nochannel = {'EMG Right','EMG Left','ECG','VEOG','HEOG'})
    %EEG = pop_eegfiltnew(EEG,8,12)
    [ALLEEG, EEG, ~] = eeg_store(ALLEEG, EEG);
end

%load data (ERP)
eeglab;
for i = 1:length(paths)
    EEG = pop_loadset(paths(i).name, paths(i).folder);
    EEG = pop_chanedit(EEG, 'lookup', 'home/cbdl/MATLAB␣Add-Ons/Collections/EEGLAB/functions/supportfiles/Standard-10-5-Cap385.sfp');
    %EEG = pop_select(EEG, nochannel = {'EMG Right','EMG Left','ECG','VEOG','HEOG'})
    %EEG = pop_eegfiltnew(EEG,8,12)
    [ALLEEG, EEG, ~] = eeg_store(ALLEEG, EEG);
end

%gfp peak aggregation (individual)
for i = 1:length(ALLEEG)
    EEG = eeg_retrieve(ALLEEG, i)
    EEG_gfp = pop_micro_selectdata(EEG, ALLEEG, 'datatype', 'spontaneous',...
    'avgref', 0, ...
    'normalise', 0, ...
    'MinPeakDist', 50, ...
    'Npeaks', 2000, ...
    'GFPthresh', 3, ...
    'dataset_idx', []);
    [ALLEEG, EEG, CURRENTSET] = eeg_store(ALLEEG, EEG_gfp, i)
end

%TAAHC clustering (individual)
for i = 1:length(ALLEEG)
    EEG = eeg_retrieve(ALLEEG, i)
    EEG_taahc = pop_micro_segment(EEG,...
    'algorithm', 'taahc', ...
    'normalise', 0, ...
    'polarity', 0);
    [ALLEEG, EEG, CURRENTSET] = eeg_store(ALLEEG, EEG_taahc, i)
end

%microstate selection
for i = 1:length(ALLEEG)
    EEG = eeg_retrieve(ALLEEG,i)
    pop_micro_plottopo(EEG)
    EEG_m = pop_micro_selectNmicro(EEG);
    [ALLEEG,~,~] = eeg_store(ALLEEG,EEG_m,i)
end

%backfit and stats
```

```
for i = 1:length(ALLEEG)
    EEG = eeg_retrieve(ALLEEG,i);
    EEG = pop_micro_fit(EEG, ...
        'polarity',0);
    EEG = pop_micro_smooth(EEG, ...
        'minTime',20, ...
        'smooth_type', 'reject␣segments', ...
        'label_type','backfit', ...
        'polarity',0);
    EEG = pop_micro_stats(EEG, ...
        'label_type','backfit', ...
        'polarity',0);
    [ALLEEG,~,~] = eeg_store(ALLEEG,EEG,i);
end

%backfit microstates to ERP
for i = 1:length(ALLEEG)
    EEG = eeg_retrieve(ALLEEG,i);
    setname = split(ALLEEG(i).setname, '/')
    subject = split(setname(end), '_')
    name = char(strcat('/home/cbdl/Downloads/Datasets/TMS_EEG/derivatives/variables/resting_microstates/alpha/',...
            regexp(ALLEEG(i).setname, '\w*[0-9]', 'match'),...
            '_ms'))
    ms = load(name)
    EEG.microstate.prototypes = ms.prototypes
    EEG = pop_micro_fit(EEG, ...
        'polarity',0);
    EEG = pop_micro_smooth(EEG, ...
        'minTime',20, ...
        'smooth_type', 'reject␣segments', ...
        'label_type','backfit', ...
        'polarity',0);
    EEG = pop_micro_stats(EEG, ...
        'label_type','backfit', ...
        'polarity',0);
    [ALLEEG,~,~] = eeg_store(ALLEEG,EEG,i);
end
```

Bibliography

Berger, H. (1929). Über das Elektrenkephalogramm des Menschen. *Archiv für Psychiatrie und Nervenkrankheiten*, *87*(1), 527–570. https://doi.org/10.1007/BF01797193

Cabral, J., Hugues, E., Sporns, O., & Deco, G. (2011). Role of local network oscillations in resting-state functional connectivity. *NeuroImage*, *57*(1), 130–139. https://doi.org/10.1016/j.neuroimage.2011.04.010

Carhart-Harris, R., Leech, R., Hellyer, P., Shanahan, M., Feilding, A., Tagliazucchi, E., Chialvo, D., & Nutt, D. (2014). The entropic brain: A theory of conscious states informed by neuroimaging research with psychedelic drugs. *Frontiers in Human Neuroscience*, *8*. Retrieved October 18, 2022, from https://www.frontiersin.org/articles/10.3389/fnhum.2014.00020

Cavanna, F., Vilas, M. G., Palmucci, M., & Tagliazucchi, E. (2018). Dynamic functional connectivity and brain metastability during altered states of consciousness. *NeuroImage*, *180*, 383–395. https://doi.org/10.1016/j.neuroimage.2017.09.065

Cohen, M. X. (2014, January 17). *Analyzing neural time series data: Theory and practice* [Google-Books-ID: EMyMEAAAQBAJ]. MIT Press.

Córdova-Palomera, A., Kaufmann, T., Persson, K., Alnæs, D., Doan, N. T., Moberget, T., Lund, M. J., Barca, M. L., Engvig, A., Brækhus, A., Engedal, K., Andreassen, O. A., Selbæk, G., & Westlye, L. T. (2017). Disrupted global metastability and static and dynamic brain connectivity across individuals in the alzheimer's disease continuum [Number: 1 Publisher: Nature Publishing Group]. *Scientific Reports*, *7*(1), 40268. https://doi.org/10.1038/srep40268

Corp, D. T., Bereznicki, H. G. K., Clark, G. M., Youssef, G. J., Fried, P. J., Jannati, A., Davies, C. B., Gomes-Osman, J., Kirkovski, M., Albein-Urios, N., Fitzgerald, P. B., Koch, G., Di Lazzaro, V., Pascual-Leone, A., Enticott, P. G., & 'Big TMS Data Collaboration'. (2021). Large-scale analysis of interindividual variability in single and paired-pulse TMS data. *Clinical Neurophysiology: Official Journal of the International Federation of Clinical Neurophysiology*, *132*(10), 2639–2653. https://doi.org/10.1016/j.clinph.2021.06.014

Deco, G., & Kringelbach, M. L. (2016). Metastability and coherence: Extending the communication through coherence hypothesis using a whole-brain computational perspective. *Trends in Neurosciences*, *39*(3), 125–135. https://doi.org/10.1016/j.tins.2016.01.001

Deco, G., Kringelbach, M. L., Jirsa, V. K., & Ritter, P. (2017). The dynamics of resting fluctuations in the brain: Metastability and its dynamical cortical core [Number: 1 Publisher: Nature Publishing Group]. *Scientific Reports*, *7*(1), 3095. https://doi.org/10.1038/s41598-017-03073-5

Eshel, N., Keller, C. J., Wu, W., Jiang, J., Mills-Finnerty, C., Huemer, J., Wright, R., Fonzo, G. A., Ichikawa, N., Carreon, D., Wong, M., Yee, A., Shpigel, E., Guo, Y., McTeague, L., Maron-Katz, A., & Etkin, A. (2020). Global connectivity and local excitability changes underlie antidepressant effects of repetitive transcranial magnetic stimulation. *Neuropsychopharmacology*, *45*(6), 1018–1025. OriginalPaper. https://doi.org/10.1038/s41386-020-0633-z

Hellyer, P. J., Scott, G., Shanahan, M., Sharp, D. J., & Leech, R. (2015). Cognitive flexibility through metastable neural dynamics is disrupted by damage to the structural connectome [Publisher: Society for Neuroscience Section: Articles]. *Journal of Neuroscience*, *35*(24), 9050–9063. https://doi.org/10.1523/JNEUROSCI.4648-14.2015

Hellyer, P. J., Shanahan, M., Scott, G., Wise, R. J. S., Sharp, D. J., & Leech, R. (2014). The control of global brain dynamics: Opposing actions of frontoparietal control and default mode networks on attention [Publisher: Society for Neuroscience Section: Articles]. *Journal of Neuroscience*, *34*(2), 451–461. https://doi.org/10.1523/JNEUROSCI.1853-13.2014

Hussain, S. (2019, February 8). *Single-pulse open-loop tms-eeg dataset* [Type: dataset]. OpenNeuro. Retrieved March 2, 2023, from https://openneuro.org/datasets/ds002094/versions/1.0.0

Jobst, B. M., Hindriks, R., Laufs, H., Tagliazucchi, E., Hahn, G., Ponce-Alvarez, A., Stevner, A. B. A., Kringelbach, M. L., & Deco, G. (2017). Increased stability and breakdown of brain effective connectivity during slow-wave sleep: Mechanistic insights from whole-brain computational modelling [Number: 1 Publisher: Nature Publishing Group]. *Scientific Reports*, *7*(1), 4634. https://doi.org/10.1038/s41598-017-04522-x

Khanna, A., Pascual-Leone, A., & Farzan, F. (2014). Reliability of resting-state microstate features in electroencephalography. *PloS One*, *9*(12), e114163. https://doi.org/10.1371/journal.pone.0114163

Klomjai, W., Katz, R., & Lackmy-Vallée, A. (2015). Basic principles of transcranial magnetic stimulation (TMS) and repetitive TMS (rTMS). *Annals of Physical and Rehabilitation Medicine*, *58*(4), 208–213. https://doi.org/10.1016/j.rehab.2015.05.005

Krzanowski, W. J., & Lai, Y. T. (1988). A criterion for determining the number of groups in a data set using sum-of-squares clustering. *Biometrics*, *44*(1), 23–34. https://doi.org/10.2307/2531893

Le Van Quyen, M. (2011). The brainweb of cross-scale interactions. *New Ideas in Psychology*, *29*(2), 57–63. https://doi.org/10.1016/j.newideapsych.2010.11.001

Lee, W. H., Doucet, G. E., Leibu, E., & Frangou, S. (2018). Resting-state network connectivity and metastability predict clinical symptoms in schizophrenia. *Schizophrenia Research*, *201*, 208–216. https://doi.org/10.1016/j.schres.2018.04.029

Lehmann, D. (1971). Multichannel topography of human alpha EEG fields. *Electroencephalography and Clinical Neurophysiology*, *31*(5), 439–449. https://doi.org/10.1016/0013-4694(71)90165-9

Lempel, A., & Ziv, J. (1986). Compression of two-dimensional data. *IEEE Transactions on Information Theory*, *32*(1), 2–8. https://doi.org/10.1109/TIT.1986.1057132

Lord, L.-D., Expert, P., Atasoy, S., Roseman, L., Rapuano, K., Lambiotte, R., Nutt, D. J., Deco, G., Carhart-Harris, R. L., Kringelbach, M. L., & Cabral, J. (2019). Dynamical exploration of the repertoire of brain networks at rest is modulated by psilocybin. *NeuroImage*, *199*, 127–142. https://doi.org/10.1016/j.neuroimage.2019.05.060

Luck, S. J., & Kappenman, E. S. (2011, December 15). *The oxford handbook of event-related potential components* [Google-Books-ID: gItoAgAAQBAJ]. Oxford University Press.

Michel, C. M., & Koenig, T. (2018). EEG microstates as a tool for studying the temporal dynamics of whole-brain neuronal networks: A review. *NeuroImage*, *180*, 577–593. https://doi.org/10.1016/j.neuroimage.2017.11.062

Pavon, J. C. H., Garces, N. S., Begnoche, J. P., Miller, L., & Raij, T. (2022, February 11). *TMS-EEG-MRI-fMRI-DWI data on paired associative stimulation and connectivity (Shirley Ryan AbilityLab, Chicago, IL)* [Type: dataset]. Openneuro. https://doi.org/10.18112/OPENNEURO.DS004024.V1.0.0

Perez, M. A., & Cohen, L. G. (2009). The corticospinal system and transcranial magnetic stimulation in stroke. *Topics in stroke rehabilitation*, *16*(4), 254–269. https://doi.org/10.1310/tsr1604-254

Plechawska-Wojcik, M., Kaczorowska, M., & Zapala, D. (2019). The artifact subspace reconstruction (ASR) for EEG signal correction. a comparative study. In J. Świątek, L. Borzemski, & Z. Wilimowska (Eds.), *Information systems architecture and technology: Proceedings of 39th international conference on information systems architecture and technology – ISAT 2018* (pp. 125–135). Springer International Publishing. https://doi.org/10.1007/978-3-319-99996-8_12

Poulsen, A. T., Pedroni, A., Langer, N., & Hansen, L. K. (2018, March 27). *Microstate EEGlab toolbox: An introductory guide* [Type: article]. bioRxiv. https://doi.org/10.1101/289850

Robertson, E. M., Théoret, H., & Pascual-Leone, A. (2003). Studies in cognition: The problems solved and created by transcranial magnetic stimulation. *Journal of Cognitive Neuroscience*, *15*(7), 948–960. https://doi.org/10.1162/089892903770007344

Sur, S., & Sinha, V. K. (2009). Event-related potential: An overview. *Industrial Psychiatry Journal*, *18*(1), 70–73. https://doi.org/10.4103/0972-6748.57865

Tanovic, E., & Joormann, J. (2019). Anticipating the unknown: The stimulus-preceding negativity is enhanced by uncertain threat. *International Journal of Psychophysiology*, *139*, 68–73. https://doi.org/10.1016/j.ijpsycho.2019.03.009

Tognoli, E., & Kelso, J. A. S. (2014). The metastable brain. *Neuron*, *81*(1), 35–48. https://doi.org/10.1016/j.neuron.2013.12.022

Van De Ville, D., Britz, J., & Michel, C. M. (2010). EEG microstate sequences in healthy humans at rest reveal scale-free dynamics. *Proceedings of the National Academy of Sciences*, *107*(42), 18179–18184. https://doi.org/10.1073/pnas.1007841107

Viesca, A., Alcauter, S., Barrios, F., González, O., & Marquez, J. (2012). Transcranial magnetic stimulation assisted by neuronavigation of magnetic resonance images. *1494*, 91–93. https://doi.org/10.1063/1.4764608

Wildie, M., & Shanahan, M. (2012). Metastability and chimera states in modular delay and pulse-coupled oscillator networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *22*(4), 043131. https://doi.org/10.1063/1.4766592

Winkler, I., Haufe, S., & Tangermann, M. (2011). Automatic classification of artifactual ICA-components for artifact removal in EEG signals. *Behavioral and brain functions: BBF*, *7*, 30. https://doi.org/10.1186/1744-9081-7-30

Wu, W., Keller, C. J., Rogasch, N. C., Longwell, P., Shpigel, E., Rolle, C. E., & Etkin, A. (2018). ARTIST: A fully automated artifact rejection algorithm for single-pulse TMS-EEG data. *Human Brain Mapping*, *39*(4), 1607–1625. https://doi.org/10.1002/hbm.23938