5-9-2017

Information Theoretic EEG Analysis of Children with Severe Disabilities in Response to Power Mobility Training: A Pilot Study

Joshua O. Usoro
Grand Valley State University

Information Theoretic EEG Analysis of Children with Severe Disabilities in Response to Power Mobility Training: A Pilot Study

Joshua Okon Usoro

A Thesis Submitted to the Graduate Faculty of

GRAND VALLEY STATE UNIVERSITY

In

Partial Fulfillment of the Requirements

For the Degree of

Master of Science in Engineering

Padnos College of Engineering and Computing

April 2017

Acknowledgements

I would like to first thank my thesis advisor Dr. Samhita Rhodes of the School of Engineering of Grand Valley State University. She has been incredibly helpful throughout the course of my studies both personally and academically, allowing me to independently realize this work while still supporting me whenever I needed guidance.

I would also like to thank the committee members, Dr. John Farris and Dr. Lisa Kenyon PT, DPT, PhD, PCS, as well as the previous graduate student Nadina Zweifel MSE, whose pioneering work made this all possible. Without their input and tireless effort in conducting the exploratory study and collecting the data on which this thesis is based on, I would not have been able to successfully complete this study.

Lastly, I would like to thank my friends and family for providing me with constant support and encouragement. I will always be eternally grateful for the love they have shown me not only during this study, but over the years. Thank you all.

Joshua Usoro

Abstract

Studies have shown that self-initiated locomotion in infants and young children is paramount to developing motor skills, social skills, and visuospatial perception. Because children with multiple, severe disabilities are often not able to produce movement on their own due to limited motor control, access to a power mobility device may be vital to their overall development. This study extends data analysis from an exploratory study, and investigates the use of information theoretics in providing objective measures to characterize changes in electroencephalograms (EEG) as a result of power mobility training, and relates these changes to observed rehabilitation outcomes. Three subjects with multiple, severe disabilities participated in the 20-week study. A single subject, A-B-A-B study design was employed where each baseline (A) and intervention (B) phase lasted 5 weeks. Various conditions were included in both phases: No Interaction 1, Passive Mobility, and No Interaction 2. EEG data were collected under each condition for 5 minutes, sampled at 128 Hz. During B phases, the subject trained for 45-60 minutes with the power mobility device as well. All three subjects showed similar changes in activation in the frontoparietal recordings, likely due to the visuomotor, attention, and memory requirements of power mobility training. The emergence of a centroparietal network was also observed in one subject. Furthermore, a theta increase and alpha decrease in power for one subject may have been the result of increased alertness and anticipation, while a decrease in theta and increase in alpha power for the other subject may be associated with an increase in cognitive performance. Additionally, all subjects demonstrated rehabilitative growth that correlated well with trends seen in the EEG metrics. All three subjects also exhibited more significant change during the intervention phases compared to the baseline phases. This suggests that power mobility training may be responsible for consistent and objectively quantifiable changes in cortical activity that may be correlated with improvement in subjective measures of cognitive gains.

Table of Contents

List of Tables

List of Figures

1 Introduction

This study was a continuation of the work performed in a previous exploratory study conducted by the Department of Physical Therapy (Dr. Lisa Kenyon), the Department of Psychology (Dr. Naomi Aldrich), the School of Engineering (Dr. John Farris and Dr. Samhita Rhodes), and the Statistics Department (Dr. Paul Stephenson) of Grand Valley State University (GVSU). The group had been working with children who have multiple, severe disabilities to improve their mobility, independence, and quality of life through the use of power mobility [1]. The group designed and built the Power Wheelchair Trainer (Trainer) [2,3], which is a motorized platform on which manual wheelchairs can be easily mounted upon to allow subjects to explore their surroundings in a safe and controlled environment. The aim of the exploratory study was to use electroencephalography (EEG) data as a physiologic measure to support the previously observed rehabilitation conclusions that power mobility training may have resulted in improvements in children's quality of life, psychosocial state, and cognitive abilities [1-3].

Findings from the exploratory study regarding the power spectral analysis of EEG data showed changes in alpha and theta power that indicated that there may have been an immediate increase in cognitive performance after undergoing power mobility training [1]. Increased cognitive ability is thought to be associated with greater interaction between areas of the brain due to the development of neural circuitry stimulated by higher levels of cortical activation [4]. This pilot study sought to investigate network connectivity within the brain as related to changes seen in the EEG power spectrum, coherence, and information theoretics.

1.1 Purpose

Self-initiated mobility is essential in typical early childhood development [5]. The ability to independently explore one's environment has been shown to improve cognitive functioning and impact a child's mental well-being and social skills [5,6]. Literature also indicates that a lack of mobility can result in a persistent pattern of learned helplessness and lack of curiosity and motivation [5]. Furthermore, restricted mobility can lead to increased dependency and inhibition [5]. In children with multiple disorders, power mobility use has been shown to have increased the number of subsequent self-initiated movements and positive peer initiations [6]. Though encouraging results have been seen, it is difficult to show large changes in power mobility skills in our specific population, just changes in basic, beginning skills. The exploratory study conducted by GVSU was the first to attempt to obtain objective physiologic data to bolster these claims by observing changes in theta and alpha power in EEG. This is only one possible measure of cognitive change however, and can be influenced by many unknown factors. The present study aims to provide an additional objective measure using the concept of neuronal network connectivity.

1.2 Scope

This study aims to identify a change in the level of information transfer and coherence between areas of the brain in children with multiple, severe disorders that have participated in power mobility training. A change in information transfer is thought to correlate with a shift in the level of preexisting network connectivity of the brain and could indicate the formation of new networks. A change in coherence could show the changing relationship between the spectra of different channels that may indicate communication between areas of the brain. These findings could help identify new measures for assessing possible emotional and cognitive changes in children with multiple developmental disorders as a result of power mobility training. While the systematic analysis of spectral content in GVSU's single-subject study elucidated changing brain activity, the relationship of these changes to neural connectivity was not explored. We expanded upon this initial study and added three new subjects to explore this possible connection. The same EEG acquisition system with minor modifications was used, as well as the same MATLAB program for signal preprocessing and spectral analysis [1]. Customized MATLAB programs were created for signal analysis of the information theoretic measures. The results of this new study may aid in ascertaining whether power mobility training influences neural network connectivity in the brain in children with multiple, severe disorders.

1.3 Assumptions

We assume that self-initiated mobility is an important component in the development of typically developing children and children with disabilities. This presumes that plasticity in a child's brain is greatly affected by the experience of movement during early childhood, rather than the mastery of the movement itself. Therefore, we believe that children with disabilities may be able to gain this experience through power mobility training, much like the learning of crawling, walking, and running observed in typically developing children [1]. Because of this, we assume that the introduction of power mobility may also have an impact on the development of the brain of a child with severe disabilities.

1.4 Hypothesis

Based on the previous work conducted by the GVSU research group, it was seen that power mobility training may have had a positive, short-term impact on cognitive performance [1]. Imaging studies in typically developing children have shown that these improvements, along with behavioral measurements, correlate to structural and anatomical changes that occur in a developing brain [7, 8]. These studies have also shown changes in neuroconnectivity, specifically in the prefrontal-parietal connectivity [7]. Therefore, we hypothesize that the cognitive impact due to power mobility training, as previously suggested, will also result in a change in connectivity and communication between areas of the brain. Because of the short-term impact previously seen, this change might occur either during or immediately after using the Trainer.

1.5 Significance

EEG spectra could be reflective of many things in the brain, including emotional state, which detracts from the significance of cognitive conclusions from previous studies. Information theoretic analysis allows for a broad, diagnostic approach to identifying cognitive impact and brain development through easily quantifiable, computationally straightforward measures. Because the use of an extracranial EEG cap occludes precise anatomical distinction, information theory allows for the measure of communication between major lobes and neuronal networks, and could form the basis for more directed, intensive cortical studies.

2 Manuscript

Abstract

Literature suggests that self-initiated locomotion in infants and young children is paramount to developing motor skills, social skills, and visuospatial perception. Because children with multiple, severe disabilities are often not able to produce movement on their own due to limited motor control, access to a power mobility device may be crucial to their overall development. This study extends data analysis from an exploratory study, and investigates the use of information theoretics in providing objective measures to characterize changes in electroencephalograms as a result of power mobility training, while relating these changes to observed rehabilitation outcomes. Three subjects with multiple, severe disabilities participated in the 20-week study. A single subject, A-B-A-B study design was employed where each baseline (A) and intervention (B) phase lasted 5 weeks. Various conditions were included during both phases: No Interaction and Passive Mobility. EEG data were collected under each condition for 5 minutes. During B phases, the subjects trained with the power mobility device as well. Transfer entropy results in all 3 subjects indicated an active frontoparietal network, likely due to the visuomotor, attention, and memory requirements of power mobility training. The emergence of a centroparietal network was also observed in one subject. A theta increase and alpha decrease in power for one subject may have been the result of increased alertness and anticipation, while a decrease in theta and increase in alpha power for the other subject may be associated with an increase in cognitive performance. All three subjects exhibited more significant change during the intervention phases compared to the baseline phases, and showed correlations between EEG metrics and rehabilitative growth, leading to the conclusion that power mobility training may be somewhat responsible for the changes seen in cortical activity and cognitive improvement.

2.1 Introduction

In typically-developing children, self-initiated mobility generally occurs between 8-16 months of age and is an integral component of growth and development [5,6,50]. The ability to move around independently and explore one’s environment has been shown to positively impact a child’s cognitive performance, motor skills, and visuospatial reasoning [5,6]. Children with severe disabilities such as cerebral palsy (CP), depending on the severity of the brain lesions and extent of the damage to the central nervous system, can exhibit posture and motor impairments that result in limited mobility. Additionally, concomitant neurological disorders to CP like seizures, sensory and perception impairments, and learning disabilities can contribute to a wide scope of afflictions resulting from CP. Children who are diagnosed with severe manifestations of CP (categorized as Level V in the Gross Motor Function Classification System, Communication Function Classification System, Eating and Drinking Ability Classification System, and Manual Ability Classification System) do not have the ability to walk or communicate basic needs, and may or may not also have cognitive limitations [51-53]. The lack of independent mobility can therefore be a significant disadvantage in the overall development of these children.

An interventional therapy known as power mobility training has been previously introduced as a method to provide self-initiated mobility to this population of children. Power mobility involves using a platform or device which equips the child with controls to move about their environment independently. Studies investigating this treatment have shown that power mobility training does indeed have a positive impact on the child’s overall well-being and disposition, however it is difficult to show large changes in power mobility skills in our specific population, just changes in basic, beginning skills. Recently, a research group at GVSU conducted an exploratory study that investigated the use of the electroencephalogram (EEG) spectrum as an objective measure to quantify changes that occurred in the brain due to power mobility training [1]. They observed a decrease in theta and increase in alpha power during the intervention phase of their study that may be associated with an immediate increase in cognitive performance due to power mobility training. They also indicated, that this could be reflective of the engagement in tasks and social interaction, as well as general, overall development over the course of the study. The exploratory study demonstrated that EEG data could be useful in providing objective measures for changes in brain activity due to power mobility training. This pilot study aims to expand upon their work by exploring changes in information theoretic measures and coherence as an indicator of functional connectivity and cortical activity within the brain due to power mobility training.

Information theory is concerned with how information is quantified, communicated, and stored within a system. In a neurological context, it allows one to examine how the nervous system communicates. Neurons are essentially information channels that are capable of relaying large quantities of information between nerve centers. Information theory characterizes random processes (e.g. EEG data) using probability density functions to estimate the amount and direction of information being transmitted within the given system. In the context of information theory, information is thought of abstractly as the resolution of uncertainty. In his paper, Claude Shannon explained this by describing information as a set of messages, where the goal was to send these messages over a noisy channel and have the receiver reconstruct these messages with a low probability of error despite the noise [15]. Therefore, when referring to the information of a random variable, the metric being alluded to is Shannon entropy. This entropy describes the reduction of uncertainty in predicting future values of that variable that occurs when one measures the current value of the variable [29]. This approach can be extended to two variables by observing the casual dependency between them. Wiener defined this dependency based on an increase of prediction power, i.e. a signal X is said to cause a signal Y when the future of signal Y is better predicted by adding knowledge from the past and present of signal X than by using the present and past of Y alone [59]. Thus, when prediction enhancement can be associated with uncertainty reduction, measures of causality can be derived from information theoretic concepts [29]. Two such concepts are mutual information and transfer entropy.

Mutual information (MI) is an information theoretic measure that detects linear and nonlinear statistical dependencies between two time series and measures how much information x, generated from signal X, provides about y, generated from signal Y, i.e. how much one random variable tell us about another. In other words, MI is a measure of how much information x gives about y, and thus is able to quantify the amount of information transmission between X and Y [27,28]. MI between completely independent signals is zero because there is not any shared information between the two time series, while higher levels of MI indicate that the amount of shared information between the signals is large. When applied to EEG data, higher levels of MI between electrodes may indicate dynamic coupling and functional connectivity between specific regions of the brain.

Transfer entropy (TE) is a related information theoretic measure that in addition to its ability to quantify the amount of coupling between variables, can also convey directional information. Unlike MI, which only measures the amount of shared information between variables, TE can indicate directional relationships between those variables which can be advantageous when analyzing physiologic systems [32]. It is similar to MI in that TE between independent signals is zero, and larger values of TE indicate a greater amount of shared information between two signals. Previous studies have used these measures to probe the structure of network firing patterns and identify hierarchical interactions in task modulations [16].

Magnitude-squared coherence (MSC), while not an information theoretic measure, is another metric that examines the relationship between two signals. MSC relates the spectral power of two time series by characterizing frequency-dependent covariance between signals, i.e. the phase coupling or linear dependency of the frequencies of two signals [67]. In regards to neuroscience, it is assumed that most cognitive processes rely on active, widespread networks within the brain that become co-activated during specific tasks. MSC has been used as a measure of this synchronous co-activation between different EEG electrode sites, as well as in individual frequency ranges such as theta and alpha, to indicate functional connectivity between areas of the brain [42]. High MSC would signify strong phase coupling between co-activated regions of the brain during various cognitive demands, suggesting the active presence of specific neuronal networks due to different tasks.

This study uses MI, TE, and MSC to investigate changes in how information is communicated within the brain due to power mobility training, and to correlate these changes with concurrently collected rehabilitation measures observed during the study.

2.2 Methods and Materials

2.2.1 Subjects

Three subjects (two male, one female) participated in this study. Parental permission was obtained for each child and the study was approved by the Human Research Review Committee at GVSU. All three subjects were diagnosed with cerebral palsy and were classified as Level V per the Gross Motor Function Classification System (GMFCS), Manual Ability Classification System (MACS), and the Eating and Drinking Ability Classification System (EDACS) [51-53]. Two of the three subjects were also diagnosed with seizures, and were classified as Level V according to the Communication Function Classification System (CFCS), whereas the third subject was classified in this system as Level IV due to his emerging communication function using an augmentative communication device. All three children used either manual wheelchairs or adaptive strollers. Subject information is summarized in Table 1.

Table 1. List of subjects and corresponding disabilities.

Participant	Age at Onset of Study	Diagnosis	GMFCS Level	MACS Level	CFCS Level	EDACS Level
1 – Female	6 years, 8 months	Spastic quadriplegic CP; Seizures	V	V	V	V
2 – Male	6 years, 4 months	Spastic quadriplegic CP; Chronic Lung Disease Seizures	V	V	V	V
3 – Male	4 years, 3 months	Athetoid CP; Congenital Heart Defect	V	V	IV	V

2.2.2 Power Mobility

A Power Wheelchair Trainer (Trainer) was designed and developed by the School of Engineering and Department of Physical Therapy at GVSU as part of the Power Mobility Project. The Trainer is a motorized platform on which any manual wheelchair can be securely mounted and temporarily converted into a power wheelchair. This allows the individual to be positioned comfortably in their own customized seating system. An individual can control the Trainer using either one button, multiple buttons (one for each direction: left, right, forward), or a joystick, that can be tailored to his or her ability level. Bumpers are affixed around the Trainer to minimize impact from collisions. For additional safety, the Trainer is also equipped with a shared control joystick that allows the supervising therapist to intervene and override movement while the subject is using the device. Throughout this manuscript, the use of the Trainer is referred to as ‘power mobility’.

2.2.3 Experimental Conditions

This study used a single-subject research design (SSRD) repeated across the three subjects. SSRDs have been used in developmental medicine and rehabilitation sciences when looking at individuals with varying degrees of multiple, severe disabilities to identify changes in target baseline variables related to an interventional treatment [54]. A non-randomized A-B-A-B design was used where A was the baseline phase and B was the intervention phase. Each phase lasted 5 weeks for a 20-week study duration. Additionally, a 6-week retention trial was conducted after the final intervention (B) phase. Training sessions occurred one day a week during baseline phases which consisted of three experimental conditions as shown in Table 2.

The intervention phase included the addition of power mobility training for 45 minutes. Power mobility used the Trainer and occurred after the passive mobility condition. Table 2 summarizes the experimental conditions. Each condition lasted 5 minutes (except for power mobility use) during which EEG data and standard video (for non-verbal behavioral cues) were continuously and simultaneously recorded. EEG data during passive and power mobility were collected despite overwhelming amounts of noise, but not used in analysis. Video was also still recorded. A 5-minute interval was chosen in order to balance our data analysis needs with realistic constraints concerning the limited attention span of the subjects [1].

Table 2. List of experimental conditions for one trial.

Number	Condition	Description
1	No Interaction 1	Sitting quietly in own wheelchair (no interaction or other stimuli)
2	Passive Mobility	Sitting in own wheelchair, playing and interacting with adults using passive mobility
3*	Power Mobility	Sitting in own wheelchair on Trainer, playing and interacting while using power mobility
1b	No Interaction 2	Sitting quietly in own wheelchair (no interaction or other stimuli)

*only during intervention phase, i.e. phase B

2.2.4 EEG Recording and Preprocessing

All preprocessing and data analysis was performed using customized software developed in MATLAB® (The Mathworks Inc., Natick, MA). EEG data were collected using a modified wireless Emotive EPOC© (Emotiv, Inc., San Francisco, CA) neuroheadset. The modified headset replaced the original system's 16 electrodes (2 reference electrodes) with gold cup disc electrodes (MVAP Medical Supplies Inc., Newbury Park, CA) that are mounted onto custom sized EasyCaps (EASYCAP GmbH, Herrsching, Germany). The electrode placement minimized artifacts seen on the outer rim and more accurately reflected electrophysiological activity occurring in the center of the scalp. The configuration used electrodes F7, F5, Fz, F4, F8, C3, Cz, C4, P7, P5, Pz, P4, P8, and Oz, referenced to AFz and FCz, according to the International 10-20 system as shown in Figure 2-1. Odd numbered electrodes are in the left hemisphere and are therefore referred to as 'left' electrodes, e.g. left parietal electrodes, while even numbered electrodes are in the right hemisphere and are similarly named, e.g. right frontal.

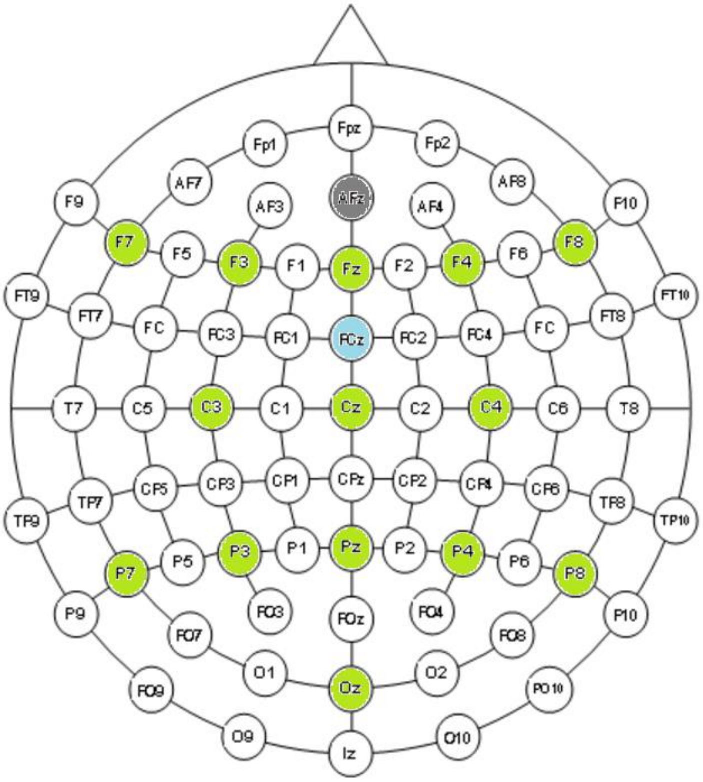


Figure 2-1. New electrode placement according to the International 10-20 system. The 16 electrodes include front (F), central (C), and parietal (P) channels, two reference electrodes (AFz & FCz). Even numbers refer to electrodes on the right hemisphere and odd numbers refer to those on the left hemisphere.

Data were recorded using Emotiv's acquisition system at a sampling rate of 2048 Hz, notch filtered at 50 Hz and 60 Hz using a digital 5th order Sinc filter. Electrode impedance was kept below 20 kΩ by applying Nuprep Skin Prep Gel at the electrode sites and dampening the subject's hair [1]. The signal is then down sampled to 128 Hz prior to being transmitted wirelessly via Bluetooth from the EasyCap to a computer, where the data were collected in TestBench™ software (Emotiv, Inc., San Francisco, CA). Video was recorded using a Microsoft LifeCam (Microsoft Corporation, Redmond, WA) to capture the subject's movements and facial expressions. EEG data and video were synchronized.

Post collection, EEG data were average referenced and filtered using a high pass 7th order Butterworth filter with a 2 Hz cut-off to remove low-frequency artifacts caused by cross-bridges. Large artifacts were manually removed while additional noise was removed using the InfoMax algorithm of wavelet enhanced independent component analysis (wICA) [1].

2.2.5 Data Analysis

2.2.5.1 Power Spectrum Density

EEG power calculation has been described in detail previously [1]. Briefly, random epochs of at least 4 seconds were chosen for power spectrum density (PSD) calculations. The PSD was calculated for all channels and normalized to the total power across all channels and frequencies using Welch's method with a Hanning window of 3 seconds and 50% overlap between segments. The frequency range from 2-12 Hz was divided into 1 Hz sub-bands (1 Hz = 4 samples with 512-point FFT), where the average power was taken across the 4 samples and normalized to the total power across channels and frequencies. Correctly identifying the different EEG rhythms, particularly the theta and alpha bands, was crucial to interpreting their functional meaning. Studies have shown that increased theta power is associated with exploratory behavior and social stimulation [39]. Therefore, theta frequency range was identified based on percent change from the No Interaction 1 to Interaction condition for each subject. Alpha range was then set to the remaining frequencies above the theta range up to 12 Hz.

2.2.5.2 Coherence

Coherence is defined as

$$C_{xy}(t,f) = \frac{S_{xy}(t,f)}{\left[S_x(t,f)S_y(t,f)\right]^{1/2}} \quad , \tag{1}$$

where S_{xy} is the cross-spectral power density and S_x and S_y are the auto-spectral power densities of *X* and *Y* respectively. The auto-spectral power densities were calculated using Welch's method with a Hamming window whose length varied based on the length of the data, a FFT length that was the next power of 2 greater than the length of the data, and a 50% overlap.

S_{xy} was computed in a similar fashion, ensemble averaging the windowed cross correlation of *X* and *Y*. The auto and cross correlation periodograms were then used to compute the magnitude-squared coherence (MSC), $C_{xy}(t,f)^2$, derived from Eq. 1. Average MSC was calculated over the theta and alpha ranges. MSC is the frequency equivalent of a correlation coefficient and provides a measure for the phase synchrony between frequencies of time series.

2.2.5.3 Mutual Information

MI was calculated between each pair of electrodes using Eq. 2 as seen below,

$$I_{XY} = \sum_{x_i, y_j} P_{XY}(x_i, y_j) \log_2 \frac{P_{XY}(x_i, y_j)}{P_X(x_i) P_Y(y_j)} \quad , \tag{2}$$

where $P_x(x_i)$ and $P_y(y_j)$ are the probabilities that an isolated measurement will find the system in the *i*th element of the histogram bin in *x* and *y* respectively, and $P_{XY}(x_i,y_i)$ is the joint probability of measurements of *X* and *Y* that produce the values *x* and *y* [27]. $P_x(x_i)$ and $P_y(y_j)$ are derived from the Shannon entropy of each system and are estimated using the histograms of *x* and *y*. Mutual information values were normalized to the maximum possible entropy between the electrodes. The histograms were created using fixed bin widths given by

$$D = \sqrt{\left(\frac{N}{5}\right)}$$

(3)

where N is the total number of points in the data set given the Cochran criterion that the number of elements in any given bin should be at least 5 for 80% of the elements [55]. Because this is the simplest method for estimating the probability density function (PDF), it is possible that the allocation of data points within the bins does not completely reflect the true distribution of the variables. In validating this approach however, varying the number of bins did not significantly change the shape of the PDF, which led to the use of Eq. 3 for optimal computational time. The fixed bin method is the least computationally demanding and performs the fastest compared to other algorithms [32]. These characteristics were ideal for our study and will be useful in future investigations with many subjects. In this scenario, mutual information will be used as more of a global indicator of shared information (regions rather than specific electrodes), where specificity is not as important.

2.2.5.4 Transfer Entropy

TE from X to Y is defined as follows:

$$T_{X\to Y} = H(y_i|y_{i-t}^{(l)}) - H(y_i|y_{i-t}^{(l)},\ x_{i-\tau}^{(k)})$$

(4.1)

$$= \sum_{y_i,y_{i-t}^{(l)}} p(y_i,\ y_{i-t}^{(l)}) \log \frac{p(y_{i-t}^{(l)})}{p(y_i, y_{i-t}^{(l)})} - \sum_{y_i,y_{i-t}^{(l)},x_{i-\tau}^{(k)}} p(y_i,\ y_{i-t}^{(l)},\ x_{i-\tau}^{(k)}) \log \frac{p(x_{i-\tau}^{(k)}, y_{i-t}^{(l)})}{p(y_i, y_{i-t}^{(l)}, x_{i-\tau}^{(k)})}$$

(4.2)

where i indicates a given point in time, τ and t are time lags for X and Y respectively, and k and l are the block lengths of past values of X and Y respectively. The past values for x and y are given as $x(k)_{i-\tau} = \{x_{i-\tau-k+1}, x_{i-\tau-k+2}, \dots , x_{i-\tau}\}$, and $y(l)_{i-t} = \{y_{i-t-l+1}, y_{i-t-l+2}, \dots , y_{i-t}\}$. Since $p(y_i, y_{i-t}^{(l)}) = \sum_{x_{i-\tau}^{(k)}} p(y_i, y_{i-t}^{(l)}, x_{i-\tau}^{(k)})$, the two arguments in the separate summations can be combined, yielding

$$T_{X\to Y} = \sum_{y_i, y_{i-t}^{(l)}, x_{i-\tau}^{(k)}} p(y_i, y_{i-t}^{(l)}, x_{i-\tau}^{(k)}) \log \frac{p(y_{i-t}^{(l)}) p(y_i, y_{i-t}^{(l)}, x_{i-\tau}^{(k)})}{p(y_i, y_{i-t}^{(l)}) p(y_{i-t}^{(l)}, x_{i-\tau}^{(k)})}$$

(4.3)

$$= \sum_{y_i, y_{i-t}^{(l)}, x_{i-\tau}^{(k)}} p\left(y_i, y_{i-t}^{(l)}, x_{i-\tau}^{(k)}\right) \log \frac{p\left(y_i | y_{i-t}^{(l)}, x_{i-\tau}^{(k)}\right)}{p\left(y_i | y_{i-t}^{(l)}\right)} .$$

(4.4)

The PDFs were estimated using the Darbellay-Vajda (D-V) adaptive partitioning method and calculated using an algorithm provided by the Harvard-MIT Division of Health Sciences and Technology [31,32]. The D-V method recursively partitions a three-dimensional space defined by series v_i, v_{i-1}, $u_{i-\tau}$ into cubes of different sizes, where v_i, v_{i-1}, $u_{i-\tau}$ are the rank-ordered time series y_i, y_{i-t}, $x_{i-\tau}$ from Eq. 4.4 respectively. The space is initially divided into 8 equal cubes where the boundaries are the mid-points in each dimension. The chi-squared statistic defined in Eq. 4.5 is used to test the null hypothesis that the data points are evenly distributed across the 8 cubes

$$s_{\chi^2} = \sum_{i=1}^{8} (M_i - \mu_M)^2 .$$

(4.5)

$M_1, M_2, \ldots, M_8$ are the number of data points contained in each cube and μ_M is the total number of data points divided by 8 [32]. If $s_{\chi^2} > \chi^2_{95\%}(7)$ (5% significance level with 7 degrees of freedom), the null hypothesis is rejected and the 8 cubes are divided into 8 smaller cubes in the same manner. The recursion continues until the null hypothesis is not rejected, in which case the current 8 cubes as a whole are taken as a single partition. The transfer entropy can then be approximated as follows:

$$T_{U\rightarrow V}(\tau) \approx \sum_{k=1}^{L} \frac{n_k}{P} \log \frac{n_k n_k^{v_{i-1}}}{n_k^{v_{i-1},u_{i-\tau}} n_k^{v_i,v_{i-1}}} , \tag{4.6}$$

where L is the number of partitions, P is the number of data triplets (v_i, v_{i-1}, $u_{i-\tau}$), n_k is the number of data points in the kth partition, and $n_k^{v_{i-1}}$, $n_k^{v_i,v_{i-1}}$, and $n_k^{u_{i-\tau},v_{i-1}}$ are the numbers of data points in the entire data that are between the bounds of the kth partition given the dimensions in the superscripts.

The D-V partitioning method was chosen over other methods (fixed-bin and kernel density estimators - KDE) due to its consistency, especially in non-linear and non-stationary processes and when data distribution is non-uniform or unknown [32,33]. It requires minimal adjusting, whereas the fixed-bin and KDE methods require choosing the number of bins and the kernel bandwidth a priori. Lee et al showed that comparisons between KDE and D-V partitioning were inconclusive, while both slightly outperformed fixed-binning [32]. Thus, D-V partitioning was chosen given its substantially shorter computational time to KDE. Fixed-binning is still computationally much faster than D-V partitioning and was used in MI estimation, where bin specificity was less crucial in providing a global view of information. Transfer entropy required more precise bin allocations to better reflect the distributions of the time-lagged series.

2.2.5.5 Rehabilitation

Various metrics were observed from a rehabilitation standpoint and tracked over the course of the study, including access method, a Wheelchair Skills Checklist, an Assessment of Learning Powered mobility use (ALP), and a Canadian Occupational Performance Measure (COPM) [60-63]. The access method refers to how the subject operated the Trainer (switch vs. joystick) and the number of switches used, i.e. the number of directions the subject was able to successfully navigate. The Wheelchair Skills Checklist details different tasks a child may perform in a motorized wheelchair and measures his or her ability to complete the task, indicating proficiency in motorized wheelchair usage. Tasks include starting and stopping on command, navigating turns, and driving in narrow areas without striking walls or objects [60]. The ALP evaluates how well an individual has learned to use power mobility and assigns a phase level based on the individual’s ability [61]. Phases range from 1 (Novice) to 8 (Expert). Lastly, the COPM is an individualized measure used to detect self-perceived change in occupational performance problems over time and has been shown to be an effective outcome measure in rehabilitation [62,63]. Because the subjects were not able to communicate their self-perceived change, the COPM was completed by their parents. Areas of perceived change include self-care, productivity, and leisure, but only measures relating to power mobility were included. All power mobility training sessions were conducted and overseen by a licensed physical therapist.

2.2.5.6 Statistical Significance

Statistical analysis was achieved using the two-standard deviation band method in which the mean of the first baseline phase is compared to the data in the subsequent phases [1]. Significance was measured by the number of data points falling outside of the ±2 standard errors from the mean of the first baseline. A significance score (s-value) ranging from 0 to ±4 (for positive and negative changes) was assigned. S-values outside the ±2 range were considered significant ($p<0.05$).

2.3 Results

Figure 2-2 shows an example PSD with a range of 0-40 Hz. The 2-12 Hz range was extracted from PSDs and used later in the identification of theta, as well as in changes in the theta and alpha bands.

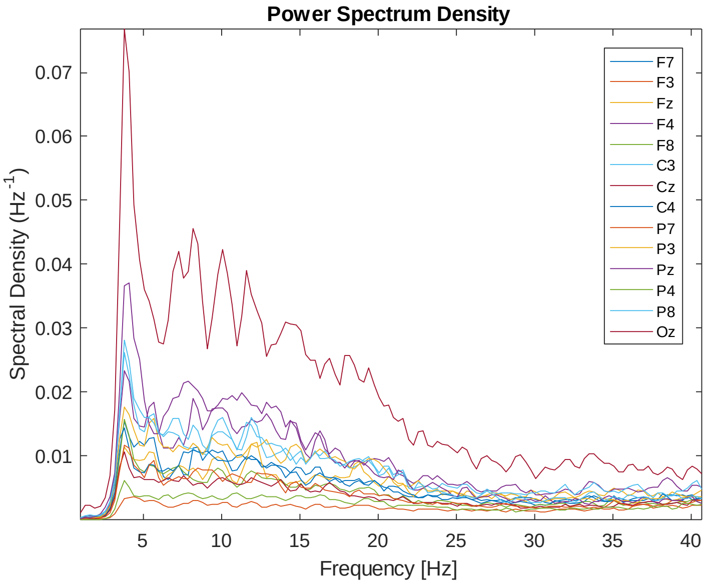


Figure 2-2. An example of a PSD plot. Spectral density is plotted as a function of frequency for each of the 14, color-coded channels.

MSC was calculated between channels as an average across both the individual theta and alpha ranges. The diagonal represents the MSC of each channel with itself, and is therefore at a maximum value. It is also a symmetric function (i.e. MSC from Channel 1 to 2 is the same as MSC from Channel 2 to 1) and this is reflected across the diagonal as seen in Figure 2-3. Coherence in the two frequency ranges of the example are very similar to one another in terms of the electrode locations of higher MSC. On closer inspection however, there are slight differences in the MSC magnitude between theta and alpha.

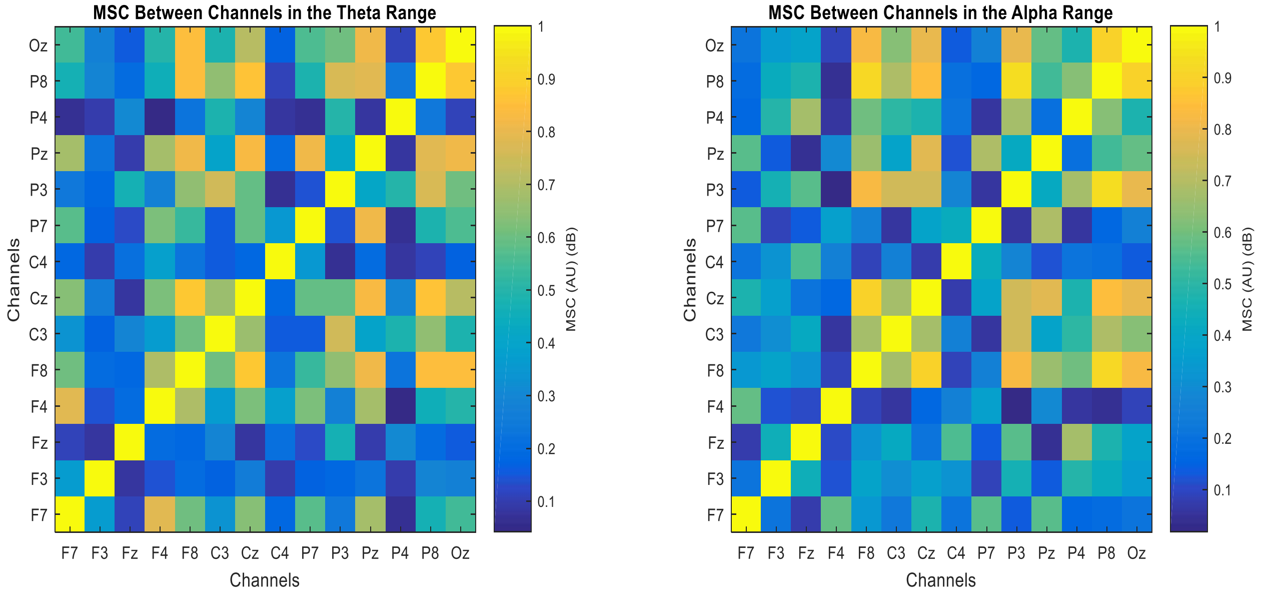


Figure 2-3. An example of a MSC plot. MSC is plotted between each channel combination. Brighter colors indicate a higher MSC value while pixels that are blue indicate lower values.

MI was calculated between each channel combination and normalized to the maximum possible amount of entropy between the two channels. An example heat map of values can be seen in Figure 2-4.

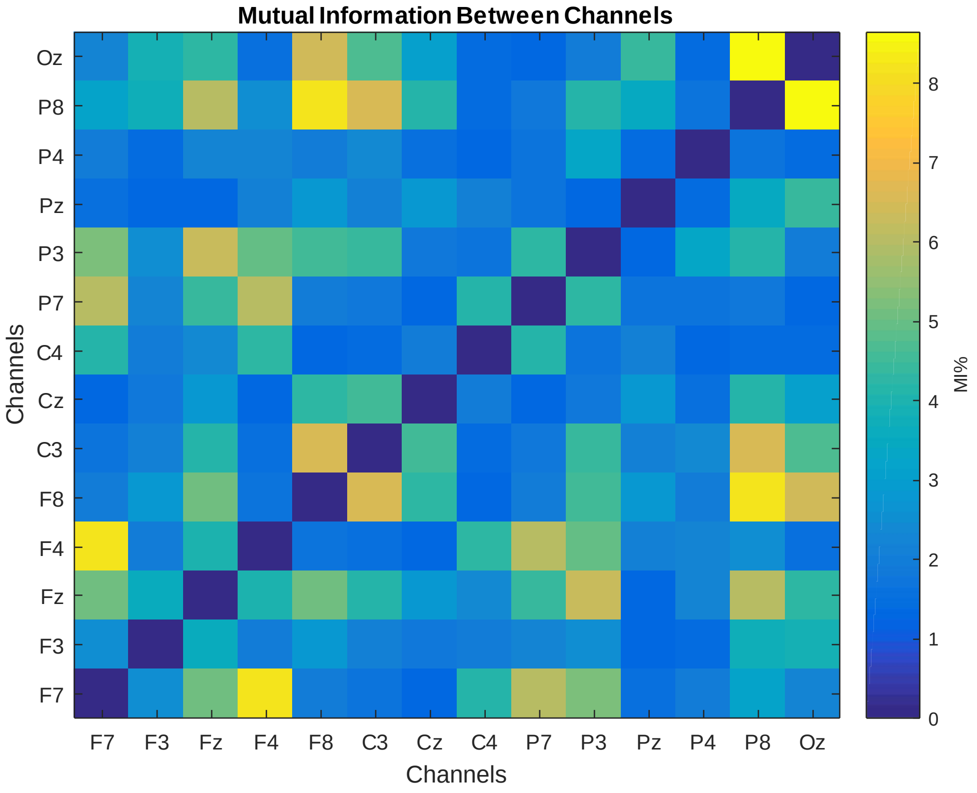


Figure 2-4. An example of a MI plot. MI is plotted between each channel combination. Brighter colors indicate a higher MI value while pixels that are blue indicate lower values.

The normalized MI is displayed as a percentage. The MI between the same electrode (dark diagonal) is the trivial result and have been set to 0 because each time series contains the same information. MI is a symmetric function, and thus Figure 2-4 is symmetric about the zeroed diagonal. These heat maps were used in later calculations to show the MI change between, and within, phases of the study.

TE was determined in a similar manner to MSC and MI, and was calculated between each combination of channels. Two methods were used to visualize TE to better depict the directional flow of information. The first method can be seen in Figure 2-5 and uses a heat map to show the

TE between channels for a single task during one data collection session. The second method displays the overall change in TE and will be seen later on.

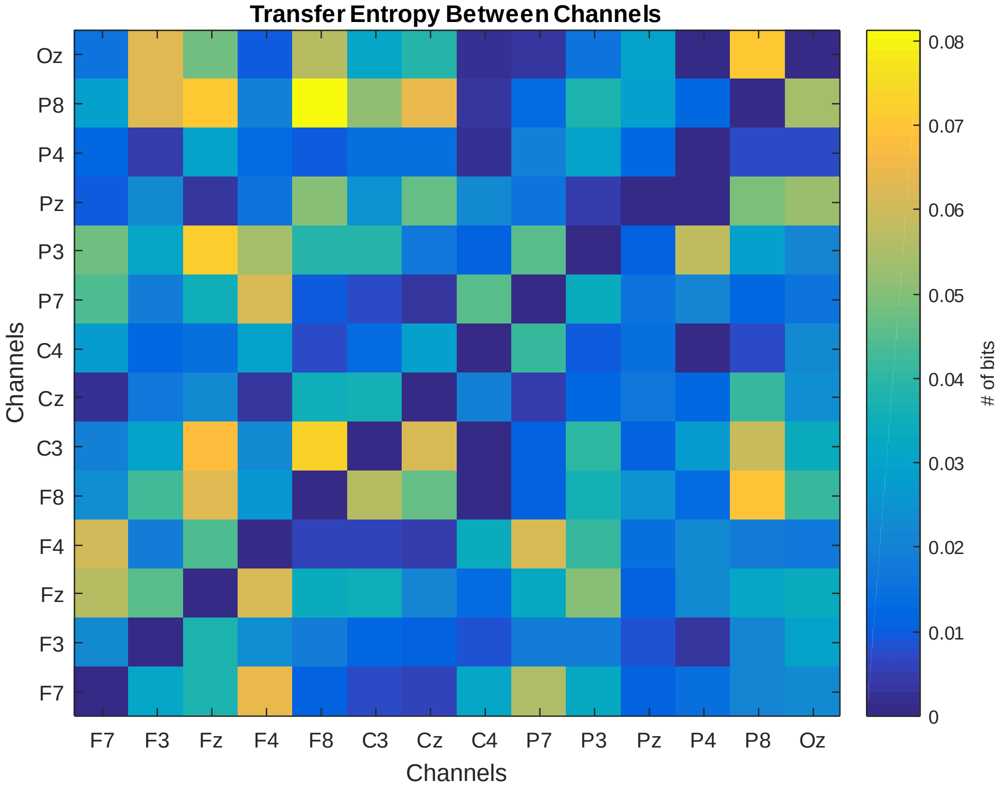


Figure 2-5. An example of a TE plot. TE is plotted between each channel combination. Brighter colors indicate a higher TE value while pixels that are blue indicate lower values. The TE plot is read from *X*→*Y*.

Like MI, the diagonal has been set to zero because the TE between the same channel is a trivial result and not useful for analysis. Unlike MI however, TE is not symmetric because the transfer of information in one direction may not be the same in the reverse direction. The heat map is read from *X* → *Y*. For example, the TE from 'F8' to 'P8' is approximately 0.08 bits.

The percent change from baseline to each phase, and within each phase, was computed for each of the aforementioned metrics, as were the significance scores. Results were separated by subject in accordance with the study design (SSRD) and can be seen in the following sections.

2.3.1 *Subject 1*

2.3.1.1 Power Spectrum Density

Theta and alpha were identified based on the plots seen in Figure 2-6. Figure 2-6a and Figure 2-6b show an increase in spectral power from No Interaction 1 to Interaction in the 2-6 Hz range.

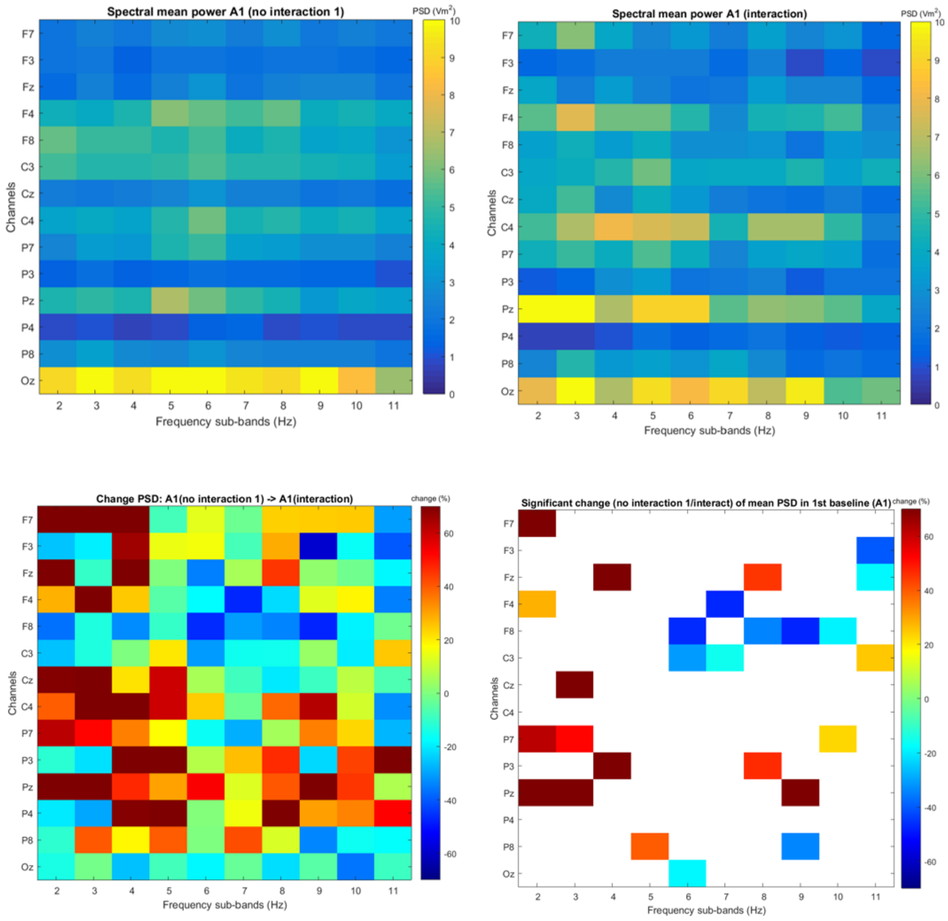


Figure 2-6 (a, b, c, d). Plots used to identify theta rhythm in Subject 1. Top left (a): spectral power during No Interaction 1. Top right (b): spectral power during Interaction. Bottom left (c): percent change in PSD. Bottom right (d): Significant percent change in PSD based on 2 standard error method.

The bottom two plots show the percent change (left) and significant change (right) from No Interaction 1 to Interaction. Significant increases can be seen in the 2-5 Hz range while areas of decrease can be seen at higher frequencies. Due to these significant changes, the theta range was identified as 2-5 Hz and the alpha range was determined to be 6-12 Hz [39].

Based on these ranges, the percent change was calculated between phases within each condition, and between conditions within each phase. Significance scores were assigned from the percent change data. The significant changes were first computed for phases within No Interaction 1 and can be seen in Figure 2-7.

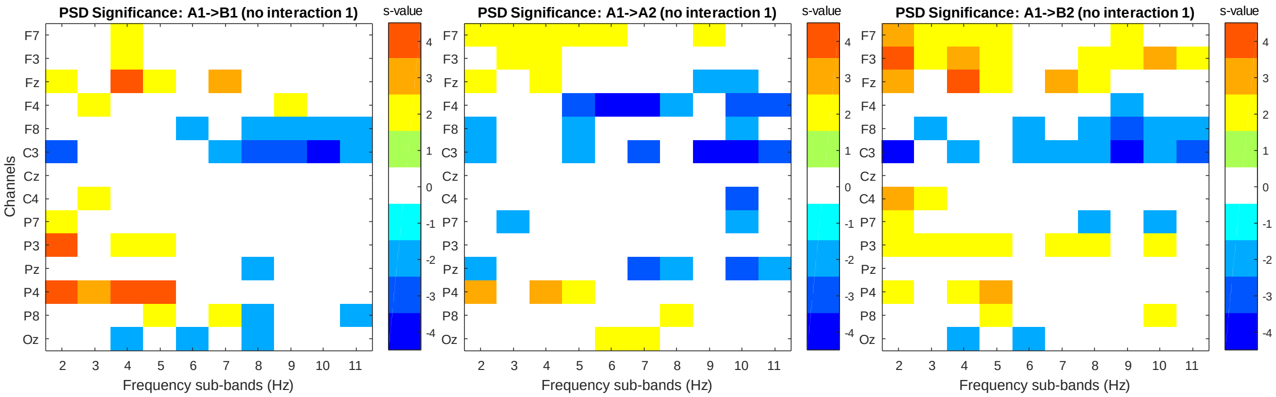


Figure 2-7. PSD significance in Subject 1 from the first baseline to each subsequent phase for the No Interaction 1 condition

Figure 2-7 shows a significant overall increase in spectral power in the theta frequency range and decrease in the alpha range for all three phases after baseline. The increase in theta power occurs almost exclusively in the parietal and frontal electrodes for all phases. Decrease in alpha power occurs in electrodes 'F8', 'C3', and 'Oz' for A1→ B1 and A1→ B2, but occurs across the frontal, central, and parietal electrodes for A1→ A2. No Interaction 2 produced a slightly different trend.

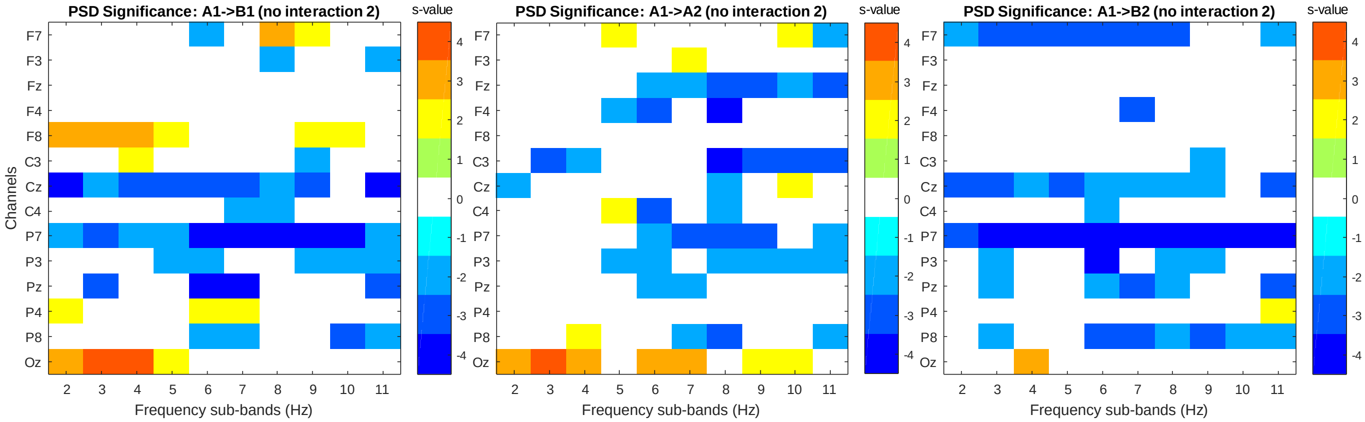


Figure 2-8. PSD significance in Subject 1 from the first baseline to each subsequent phase for the No Interaction 2 condition

Change from A1→ B1 shows a rise in theta power for 'F8' and 'Oz' and a suppression of alpha in the right parietal hemisphere. Additionally, broadband reduction in PSD was seen in the left parietal hemisphere and central midline ('Cz'). For A1→ A2, the occipital midline showed broadband increase in power while the frontal, central, and parietal electrodes all decreased in spectral power. Long term change from A1→ B2 showed broadband decreases in power the central and parietal electrodes, as well as 'F7'. There was no long-term trend of increasing power for the No Interaction 2 condition. The last condition examined for PSD was the change within each phase between No Interaction 1 and No Interaction 2.

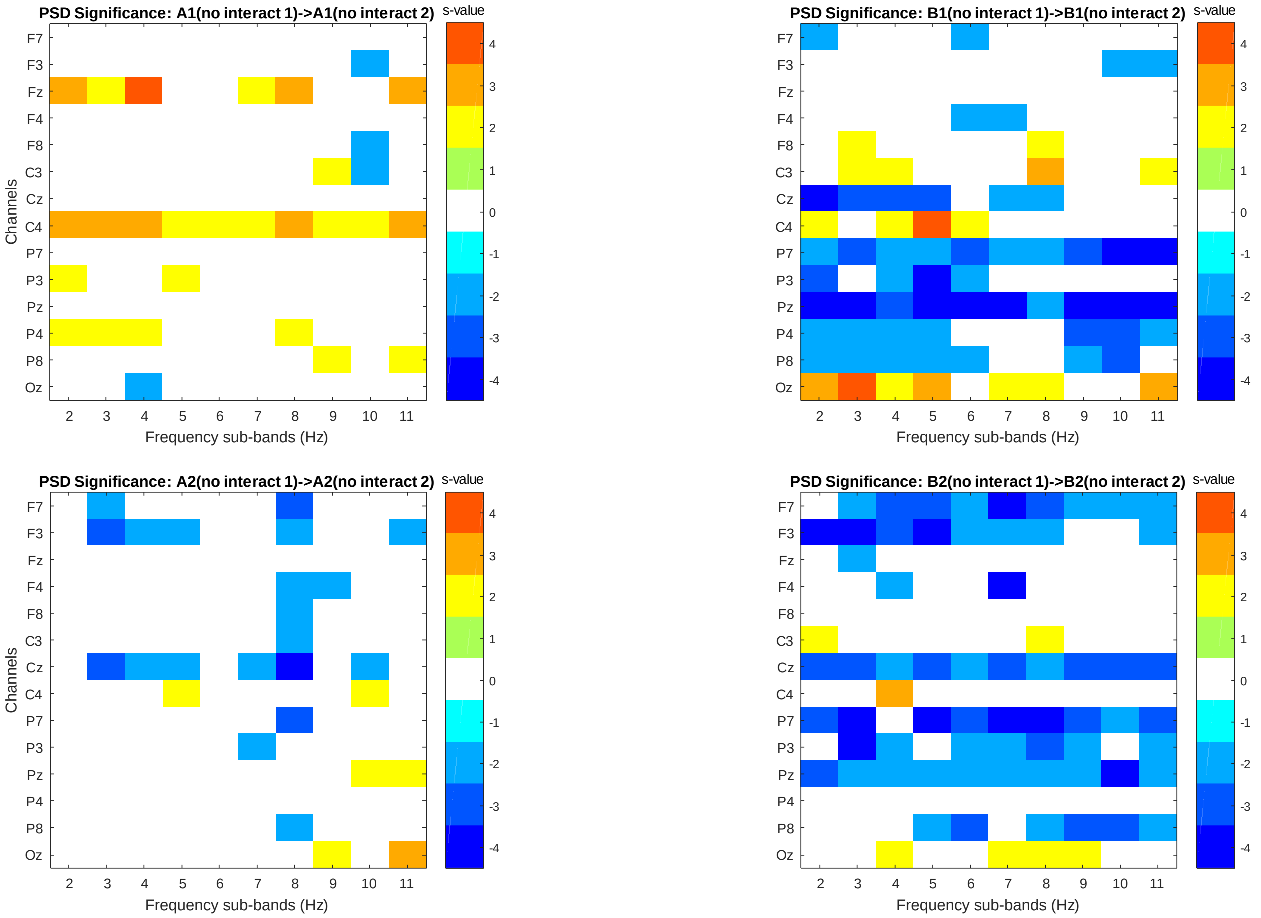
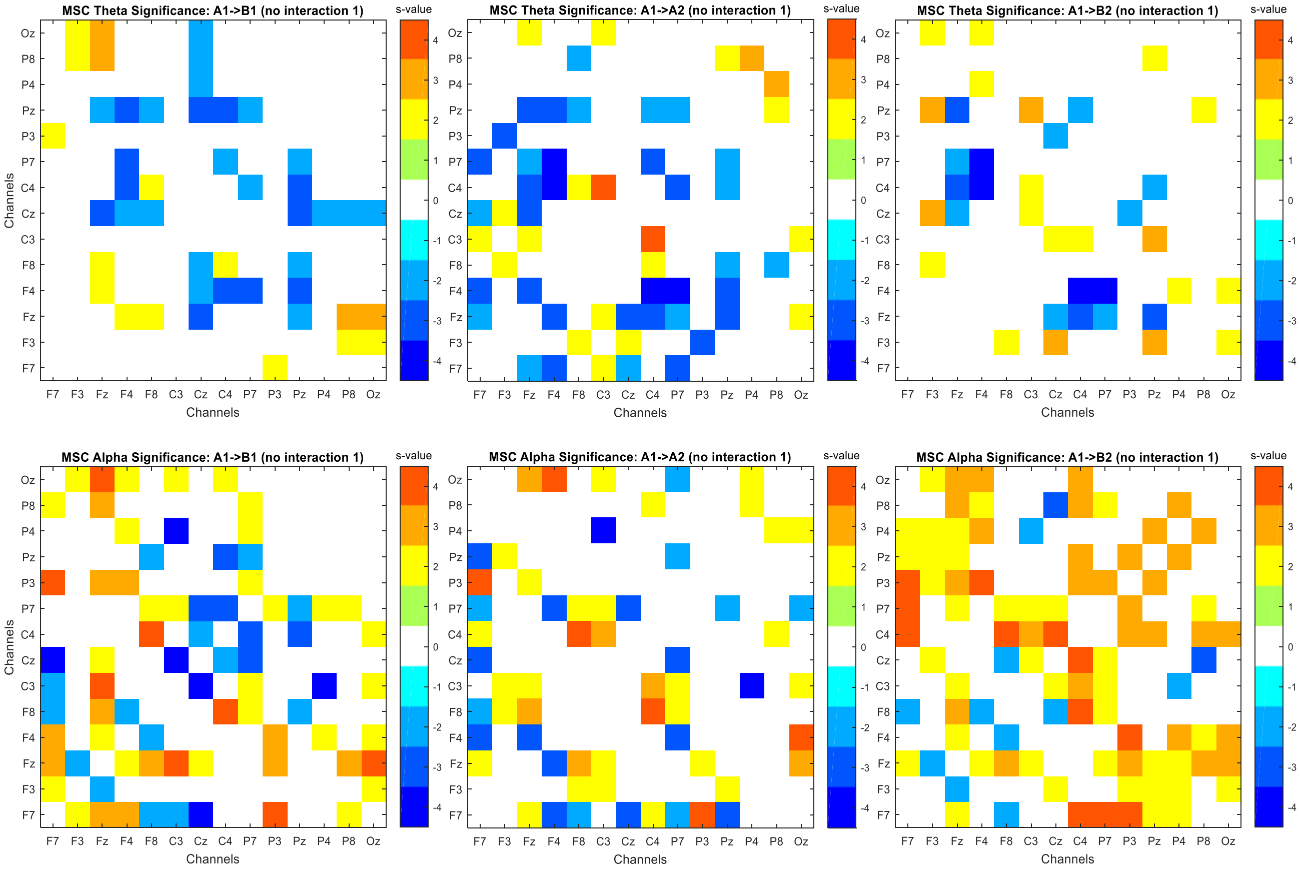


Figure 2-9. PSD significance in Subject 1 within each phase from the No Interaction 1 to the No Interaction condition

Figure 2-9 shows that overall, more significant change occurred during intervention (B) phases than during baseline phases (A). Little change was exhibited during A1 (top left) except for broadband spectral increase in the frontal midline ('Fz') and 'C4', while a minor amount of power reduction occurred during A2 at 8 Hz and in the central midline and left frontal lobe. Both intervention phases produced significant broadband suppression in the parietal lobe and central midline. A decrease was also seen in the frontal lobe, mainly in the left hemisphere, during B2. Spectral power rose in 'C3', 'C4', and 'Oz' for both intervention phases as well.

2.3.1.2 Magnitude-Squared Coherence

Visualization of MSC also involved heat maps to display significance between electrode pairs. Changes were examined between and within phases as well. MSC changes in the theta range can be seen in the top row of the figures while the alpha range is displayed on the bottom.


MSC Theta Significance: A1->B1 (no interaction 1)

MSC Theta Significance: A1->A2 (no interaction 1)

MSC Theta Significance: A1->B2 (no interaction 1)

MSC Alpha Significance: A1->B1 (no interaction 1)

MSC Alpha Significance: A1->A2 (no interaction 1)

MSC Alpha Significance: A1->B2 (no interaction 1)

Figure 2-10. MSC significance in Subject 1 from the first baseline to each subsequent phase for the No Interaction 1 condition. MSC within the theta range is seen within the top 3 plots while MSC in the alpha range is seen in the bottom 3 plots.

More significant change was exhibited overall in the alpha range than in the theta range during No Interaction 1. MSC between the frontal and parietal lobes, as well as between the right central hemisphere and frontal lobe, appeared to diminish in the theta range over time. The alpha range depicted the opposite trend, where MSC between the frontal and parietal lobes increased over time. MSC within the central electrodes and within the parietal electrodes both rose as well during B2 in the alpha range. No Interaction 2 indicated trends that were similar to those seen in Figure 2-10. MSC decreased overall in both frequency ranges from A1→ B1 and from A1→ B2. This decrease occurred in seemingly random electrode pairs rather than in concentrated regions. A1→ A2 showed an overall increase in MSC, especially in the theta range. Increases could be seen between central and frontal electrodes, frontal and parietal electrodes, and within parietal electrodes.

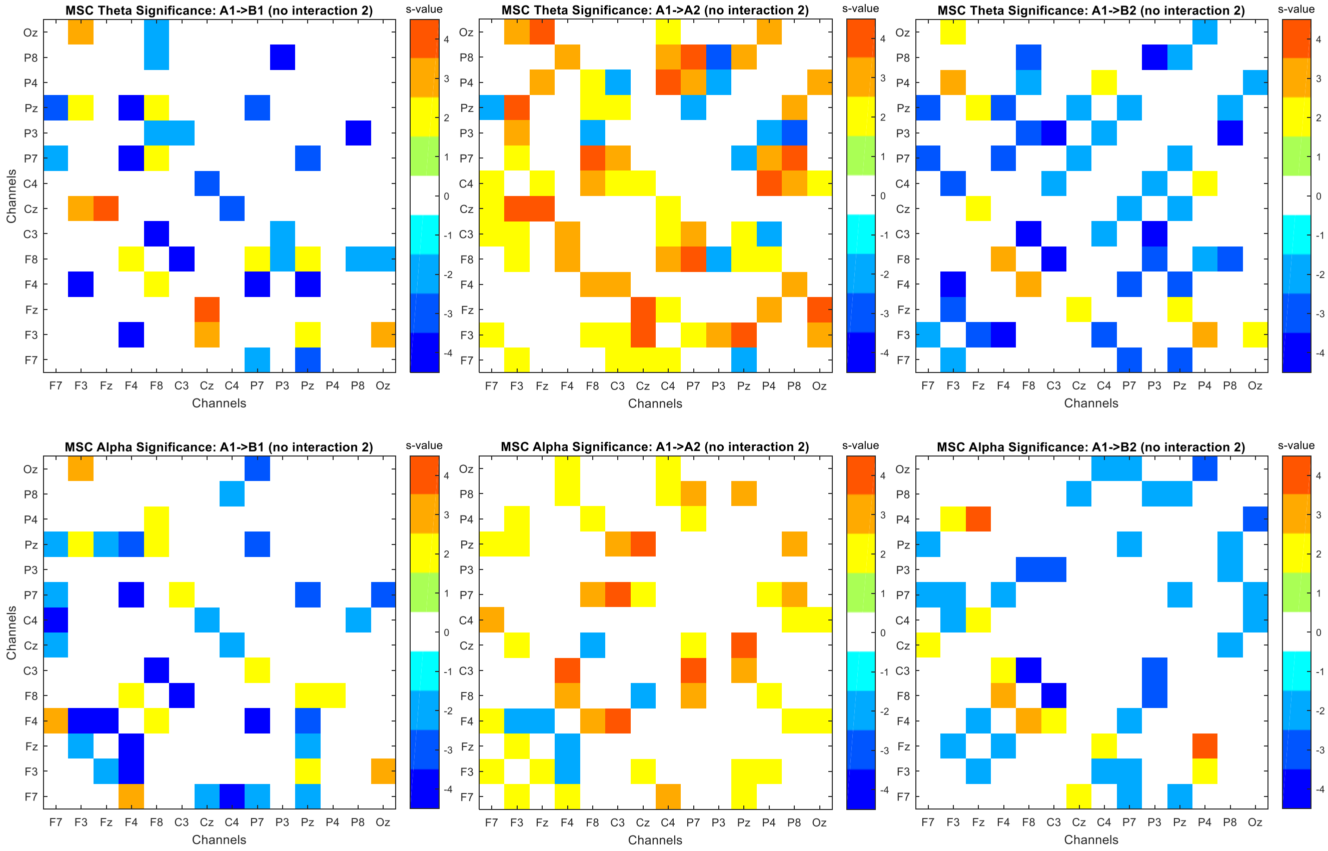


Figure 2-11. MSC significance in Subject 1 from the first baseline to each subsequent phase for the No Interaction 2 condition. MSC within the theta range is seen within the top 3 plots while MSC in the alpha range is seen in the bottom 3 plots.

Lastly, MSC was examined within each phase for both frequency ranges. ‘No Interaction 1’, ‘No Interaction 2’, and ‘s-value’ were abbreviated as ‘NI 1’, ‘NI 2’, and ‘s’ respectively in Figure 2-12. In the theta range, A1, B1, and B2 all show a general reduction in MSC between the frontal lobe and all other electrodes, while A2 shows a rise between these same electrodes. During phase A2 there is an additional increase in MSC within the parietal electrodes. The alpha frequency range shows a similar trend in significant change but to a lesser extent. MSC change was not concentrated in specific electrode regions but was spread out sporadically among electrode pairs.

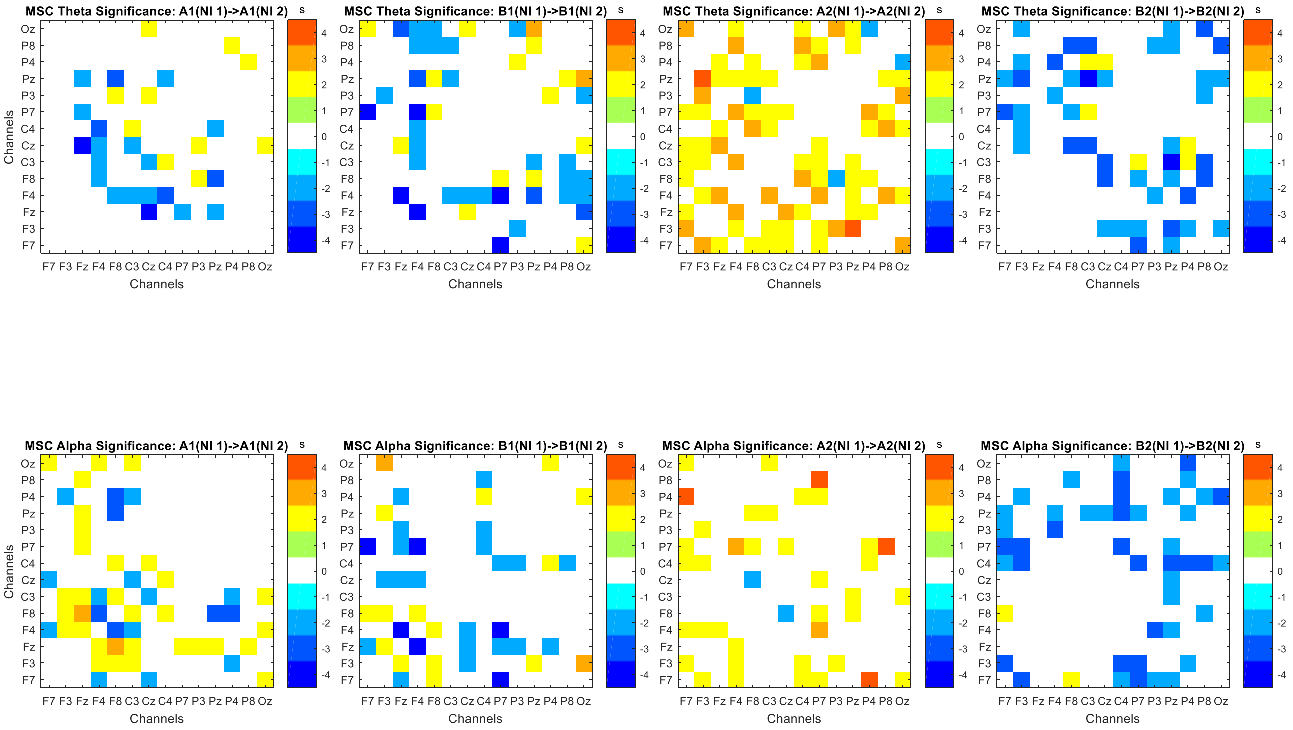


Figure 2-12. MSC significance in Subject 1 within each phase from the No Interaction 1 to No Interaction 2 condition. MSC within the theta range is seen within the top 4 plots while MSC in the alpha range is seen in the bottom 4 plots.

2.3.1.3 Mutual Information

Significant change in MI was calculated and visualized in a manner similar to PSD. Figure 2-13 shows significant changes between electrodes for the No Interaction 1 condition.

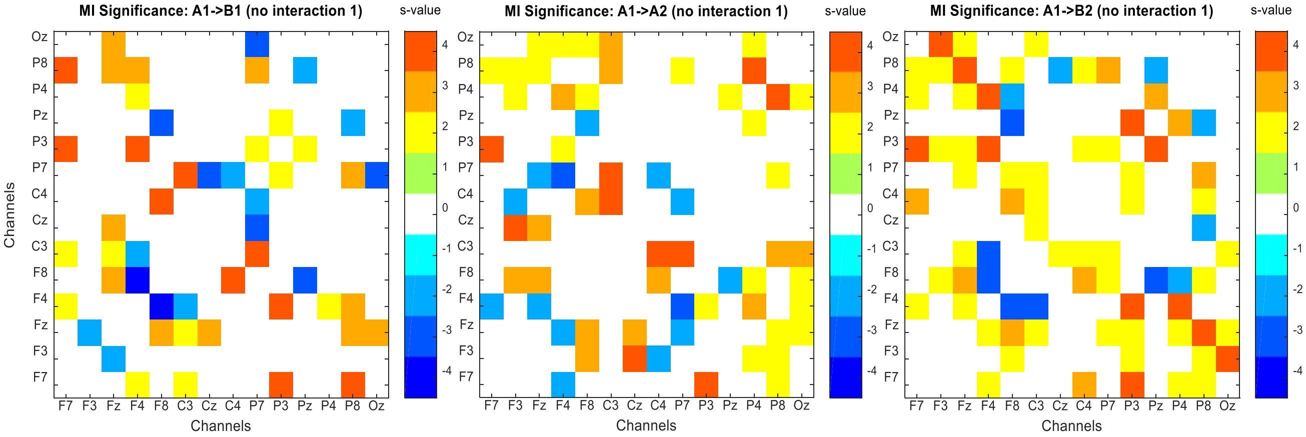


Figure 2-13. MI significance in Subject 1 from the first baseline to each subsequent phase for the No Interaction 1 condition

MI is a symmetric function so values are reflected about the diagonal axis. Overall, all 3 phases show an increase in MI. The plots for phases A2 and B2 show an interesting trend in that there is a cluster of significant increase in MI between the frontal and parietal lobes. These were the same lobes that often showed significant change in PSD as well. No Interaction 2 produced results for MI that were similar to that of PSD. Figure 2-14 shows a decreasing trend in MI for B1 and B2. A2 displayed overall higher levels of MI compared to baseline.

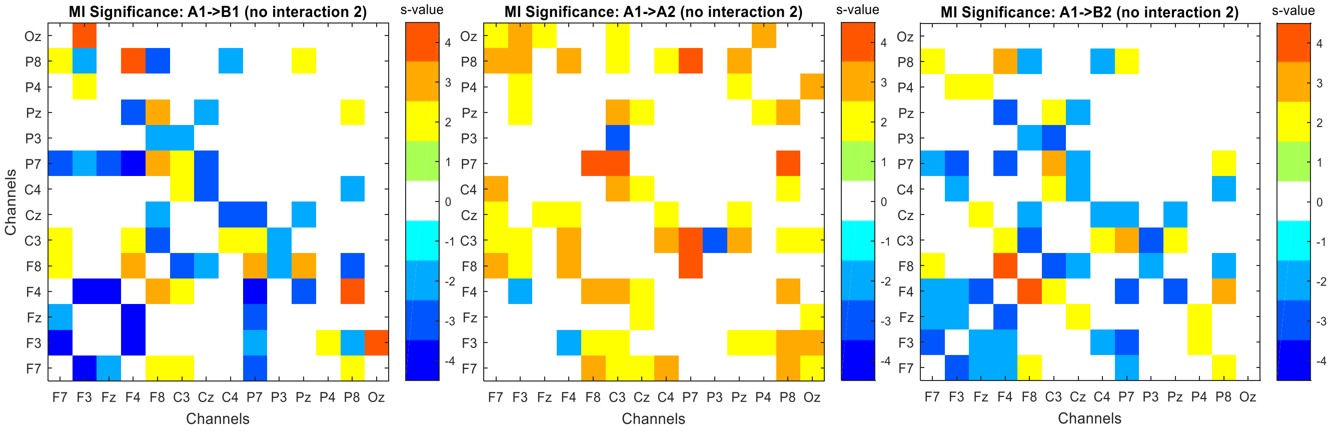


Figure 2-14. MI significance in Subject 1 from the first baseline to each subsequent phase for the No Interaction 2 condition

A reduction in MI can be seen for electrode pairs within the frontal lobe and sparingly among electrode pairs between the frontal lobe, parietal lobe, and central midline in B2. Phase A2 indicates regions of MI increase between frontal and parietal lobes, and well as between frontal and central lobes. Analysis within the Interaction condition were examined, but little change was seen. This was common throughout the metrics included in this study. Lastly, MI was examined within each phase. Again, much more significant change was seen during the intervention phases than during baseline phases. B1 and B2 both show significant decrease among the frontal electrodes and between the frontal and parietal electrodes.

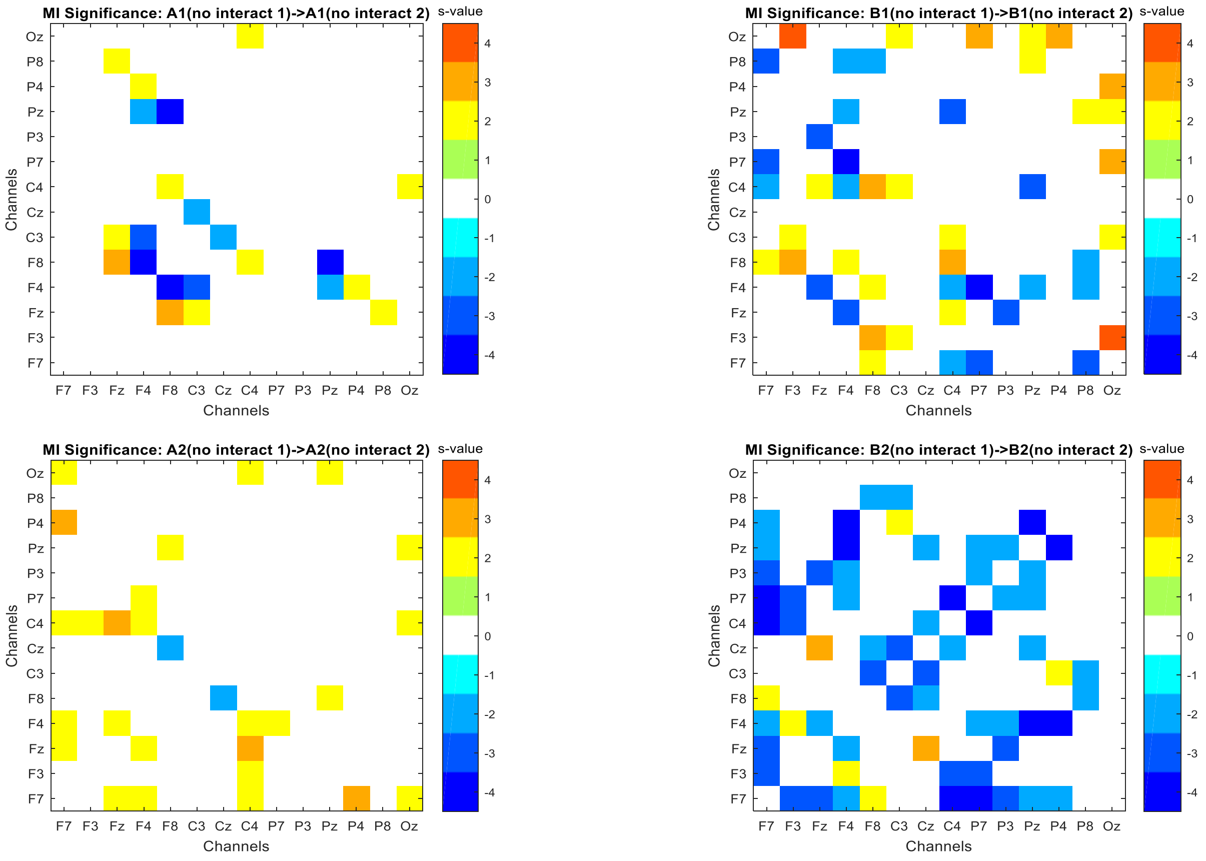


Figure 2-15. MSC significance in Subject 1 within each phase from the No Interaction 1 to No Interaction 2 condition.

2.3.1.4 Transfer Entropy

TE was visualized in two different manners. The first method involved using heat maps like those that were used in previous sections. Because TE has directionality though, a second technique was used to provide plots that served as additional reference to the direction that information was flowing. TE in the heat maps is read from $X \rightarrow Y$, while arrows indicate direction in the electrode maps.

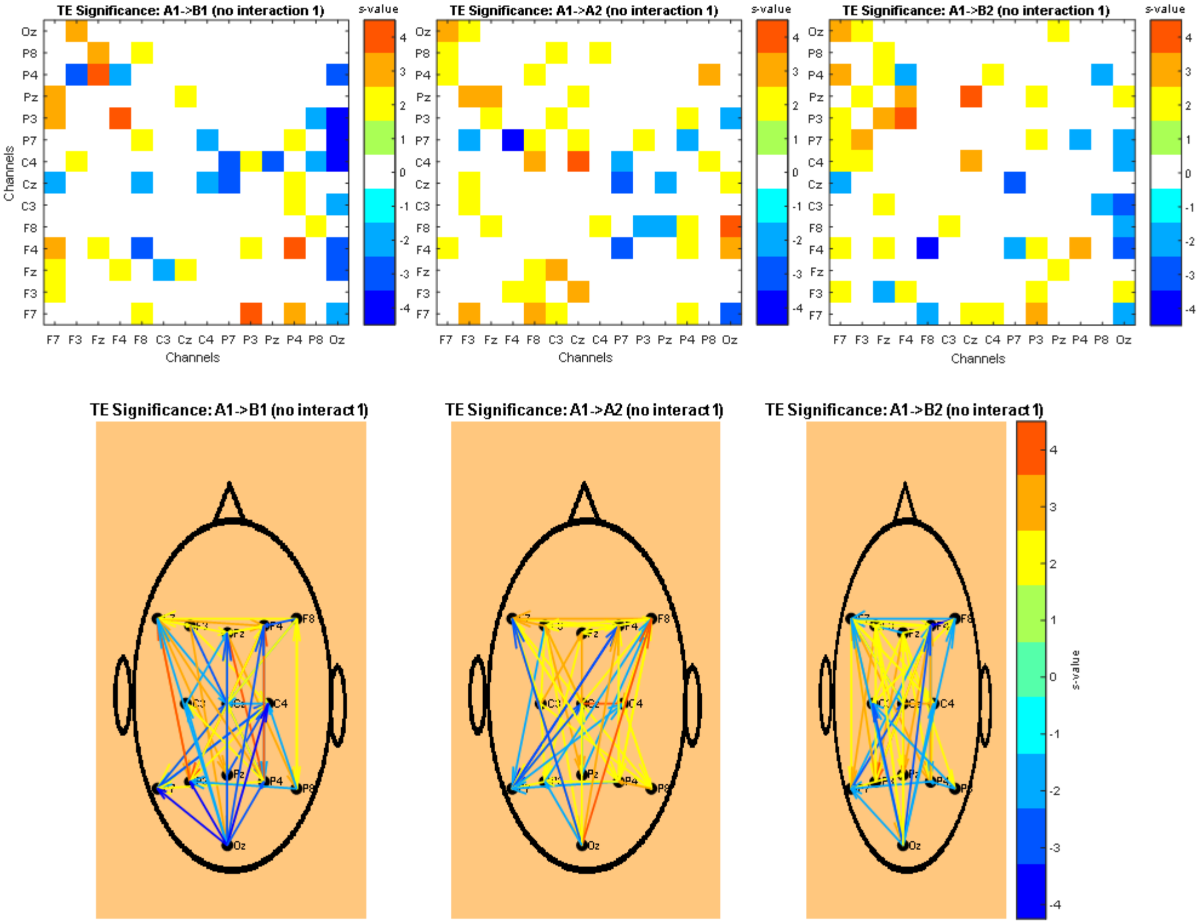
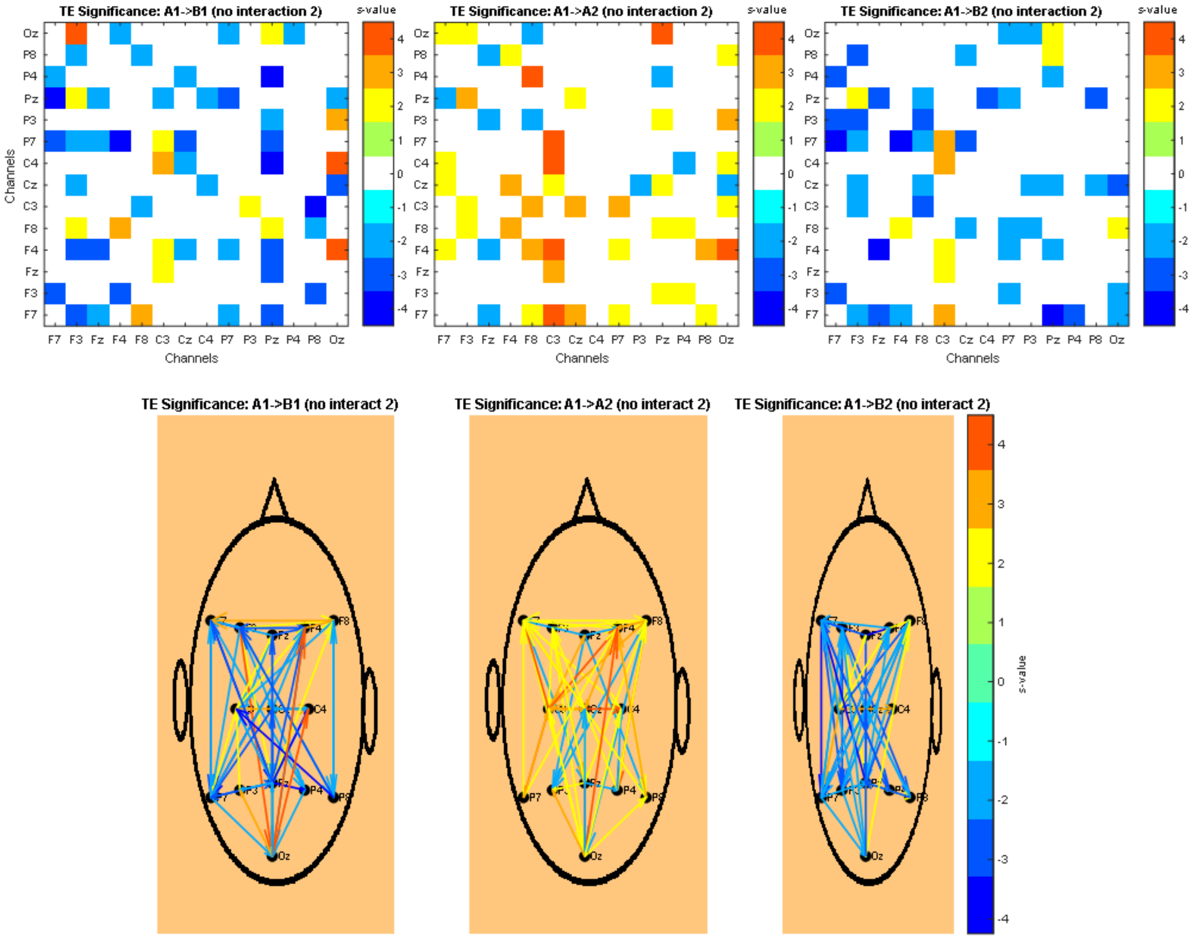


Figure 2-16. TE significance in Subject 1 from the first baseline to each subsequent phase for the No Interaction 1 condition. The top 3 plots depict TE in a heat map with the direction of information moving from *X*→*Y*. The bottom 3 plots also show TE using arrows from one electrode to the next.

Figure 2-16 shows that during the No Interaction 1 condition, there was a consistent rise in TE from the frontal to parietal lobes, and an overall decline in TE from the occipital midline to all other electrodes. TE analysis during No Interaction 2 produced plots in line with the other metrics. Figure 2-17 indicates that B1 and B2 caused a general decreasing trend in TE, particularly from frontal to parietal lobes during B2. Conversely, A2 showed significant increase in TE for most electrode pairs, especially from the central electrodes to others.


TE Significance: A1->B1 (no interaction 2)

TE Significance: A1->A2 (no interaction 2)

TE Significance: A1->B2 (no interaction 2)

TE Significance: A1->B1 (no interact 2)

TE Significance: A1->A2 (no interact 2)

TE Significance: A1->B2 (no interact 2)

Figure 2-17. TE significance in Subject 1 from the first baseline to each subsequent phase for the No Interaction 2 condition. The top 3 plots depict TE in a heat map with the direction of information moving from *X*→*Y*. The bottom 3 plots also show TE using arrows from one electrode to the next.

Finally, TE was examined within each of the phases. As was seen with other metrics, baseline phases produced significantly less change than did intervention phases. TE did not show any trends during baseline but decreased during intervention phases between the frontal and parietal lobes, as well as between parietal electrodes (B2).

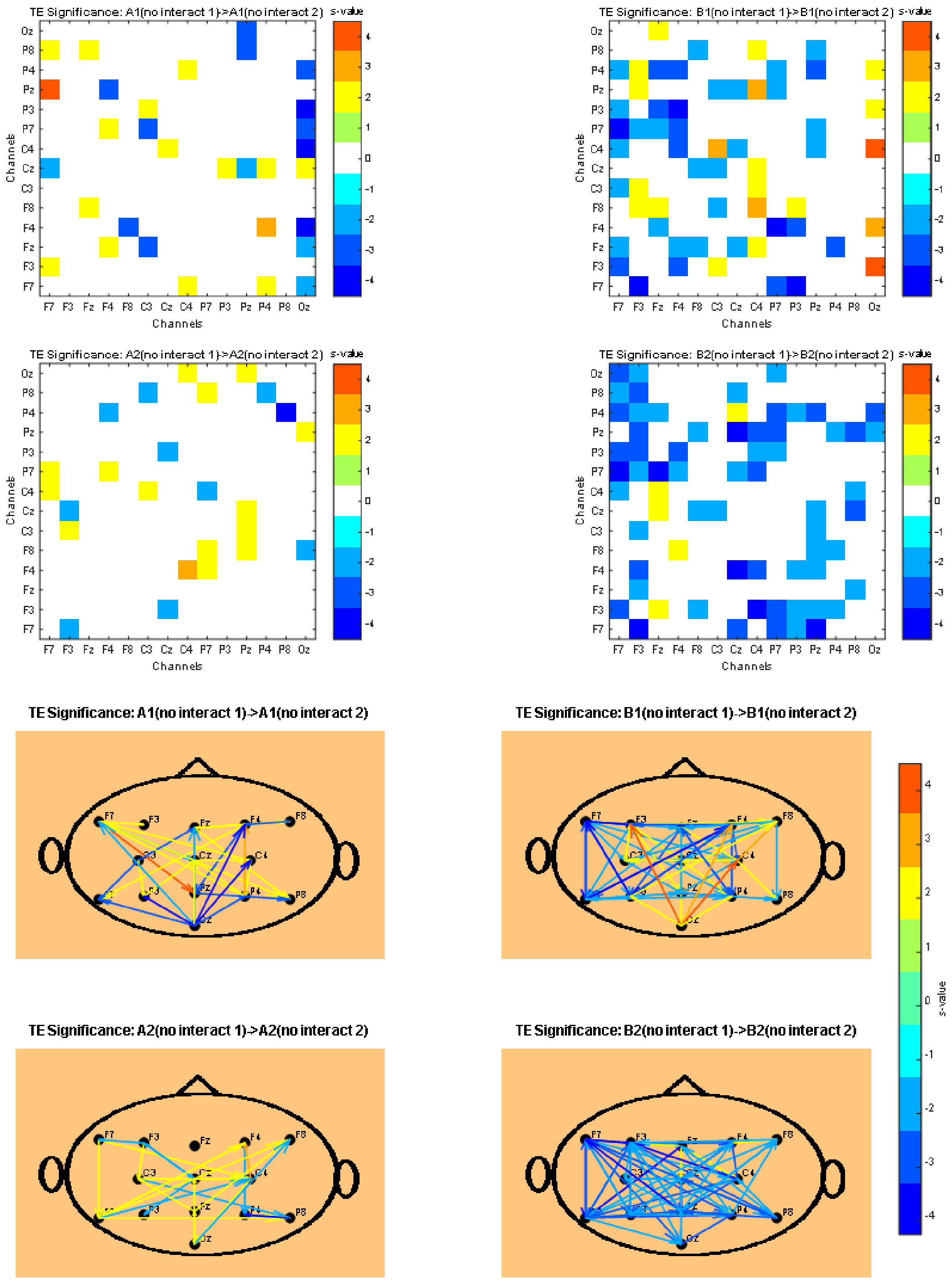


Figure 2-18. TE significance in Subject 1 within each phase from the No Interaction 1 to No Interaction 2 condition. The top 4 plots depict TE in a heat map with the direction of information moving from $X \rightarrow Y$. The bottom 4 plots also show TE using arrows from one electrode to the next.

2.3.2 Subject 2

2.3.2.1 Power Spectrum Density

Figures 2-19a and 2-19b show a rise in spectral power in the 2-5 Hz range and little difference in spectral power for frequencies above that. Significant increases in Figures 2-19c and 2-19d can also be seen in the 2-5 Hz while areas of either decrease or no significant change can be seen at higher frequencies. Due to these findings, the theta range was identified as 2-5 Hz and the alpha range was determined to be 6-11 Hz.

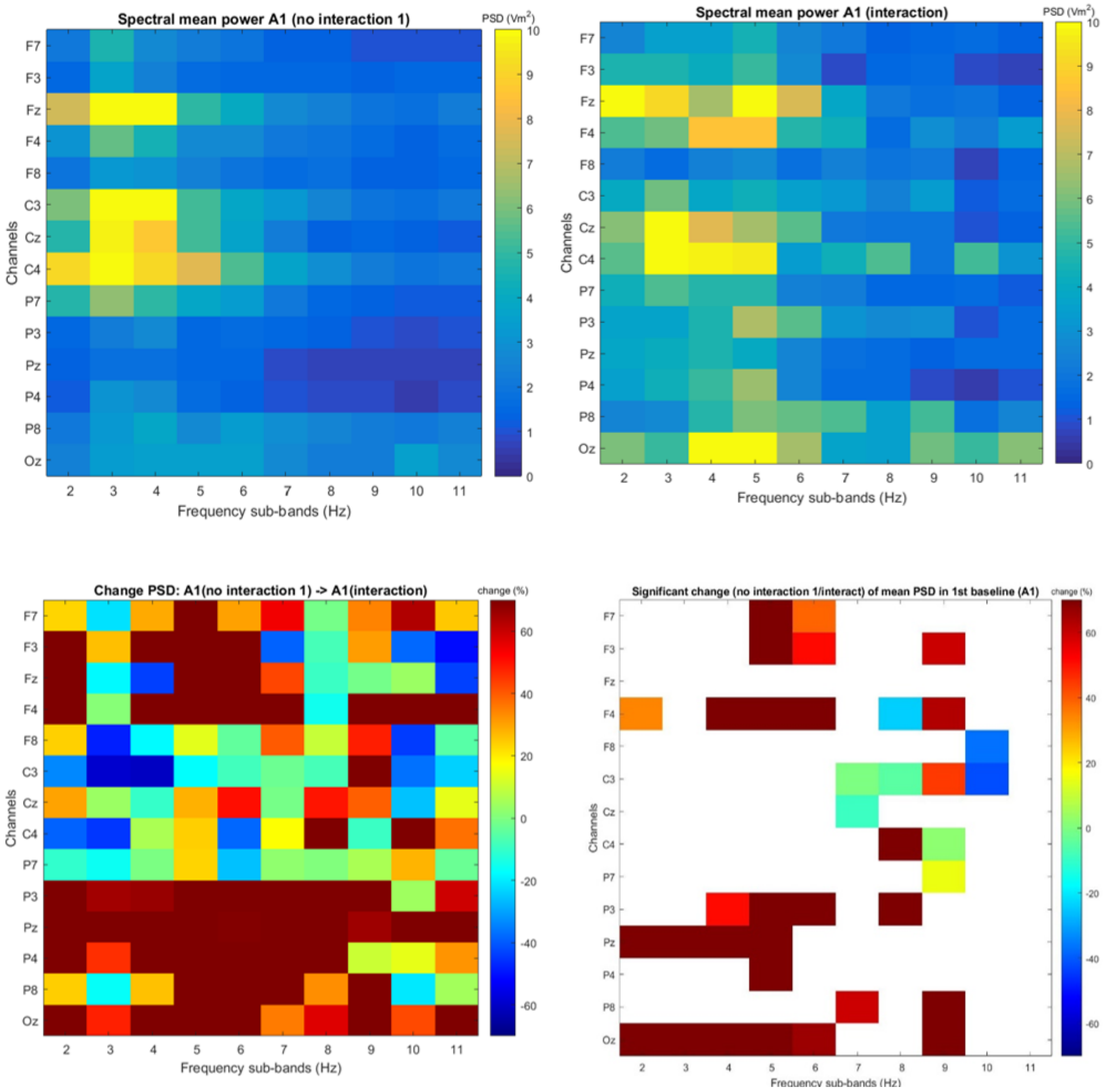


Figure 2-19 (a, b, c, d). Plots used to identify theta rhythm in Subject 2. Top left (a): spectral power during No Interaction 1. Top right (b): spectral power during Interaction. Bottom left (c): percent change in PSD. Bottom right (d): Significant percent change in PSD based on 2 standard error method.

Figure 2-20 shows increases in both the theta and alpha ranges for all 3 phases following the first baseline. A2 and B2 shows a minor amount of scattered decrease in alpha power, but long term (A1→B2), there appears to be broadband increase in spectral power.

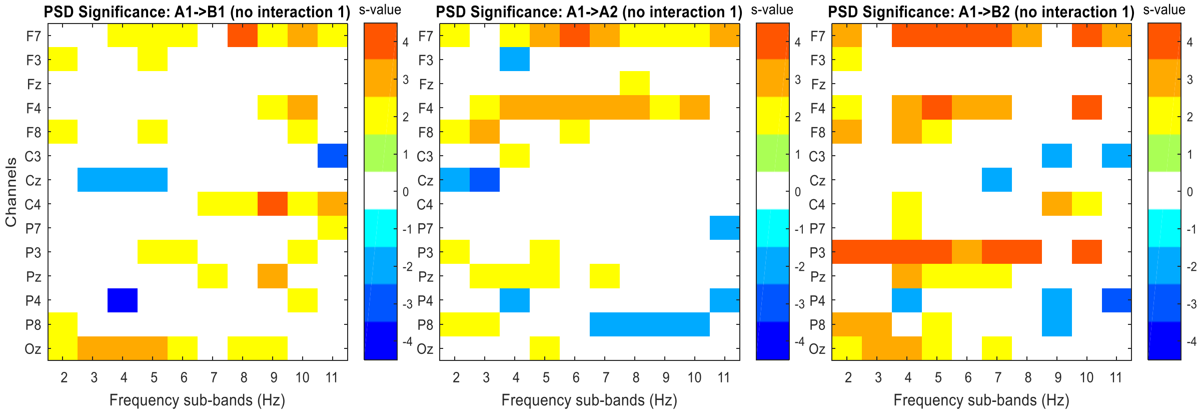


Figure 2-20. PSD significance in Subject 2 from the first baseline to each subsequent phase for the No Interaction 1 condition

No Interaction 2 produced a slightly different trend. B1 shows a decrease in theta power and an increase in alpha power. A2 caused a general rise in spectral power, mostly concentrated in the frontal lobe. A1→ B2 exhibited a trend similar to B1 in that there was more suppression in PSD occurring in the theta range while increases were seen in the alpha range.

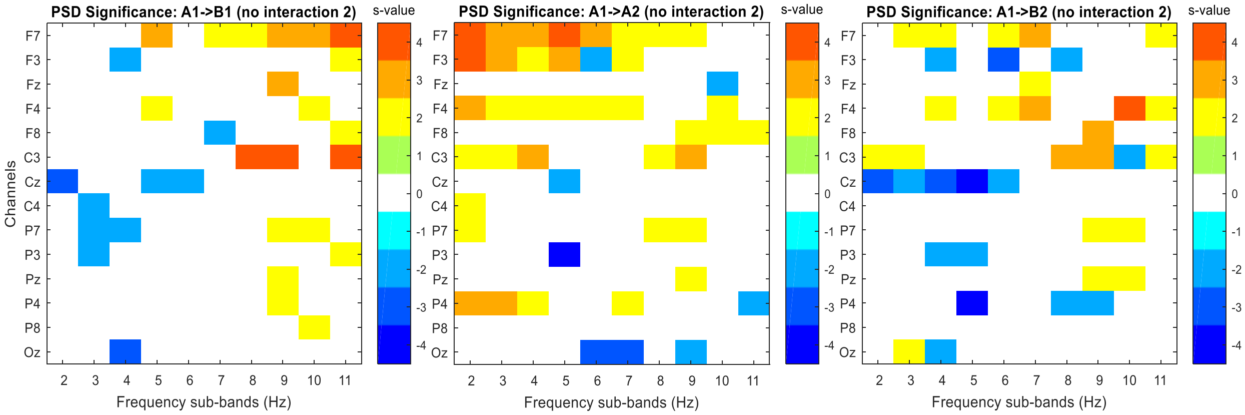


Figure 2-21. PSD significance in Subject 2 from the first baseline to each subsequent phase for the No Interaction 2 condition

The last condition examined for PSD was the change within each phase between No Interaction 1 and No Interaction 2. Figure 2-22 shows that like Subject 1, slightly more significant change occurred overall during intervention (B) phases than during baseline phases (A). Baseline phases generally caused a broadband increase in spectral power while intervention phases caused a decline. B2 indicates a decrease in theta power in the frontal electrodes, an increase in alpha power in the right frontal and left central hemisphere electrodes, and a broadband decrease in power for the right central and left parietal electrodes.

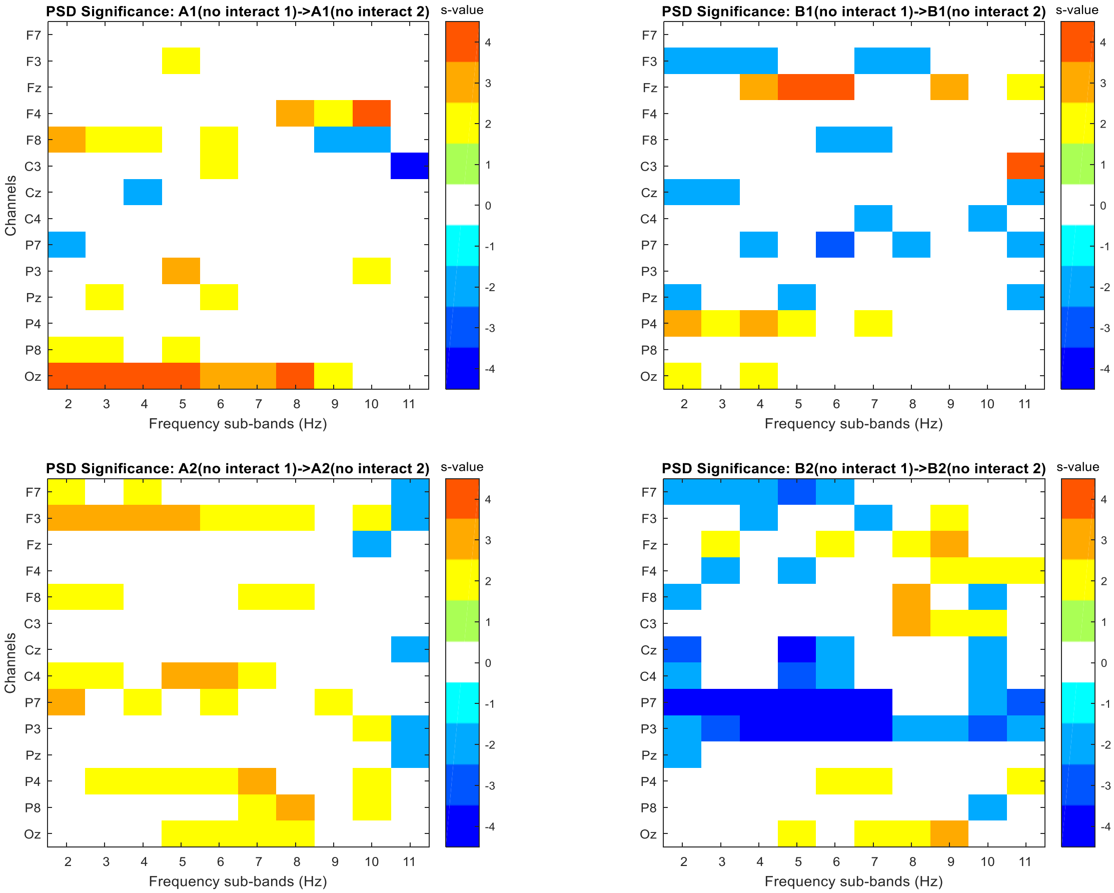


Figure 2-22. PSD significance in Subject 2 within each phase from the No Interaction 1 to No Interaction 2 condition

2.3.2.2 Magnitude-Squared Coherence

More significant change was exhibited overall in the alpha range than in the theta range during No Interaction 1. MSC in theta does not appear to significantly change as a result of the B1 and A2 phases. There is however, a rise in coherence between the central and frontal electrodes during B2. The alpha range after B1 depicted a cluster of increased MSC between the parietal electrodes in the left hemisphere and the central electrodes. A2 seems to have caused an elevation in MSC between the parietal and central electrodes, as well as within the parietal lobe. Long term, there appears to be a rise in coherence between the parietal and central electrodes, parietal and frontal electrodes, and within both the frontal and parietal lobes, as seen in alpha A1→ B2. It is interesting to note the reduction in MSC associated with the parietal midline ('Pz') in this plot as well.

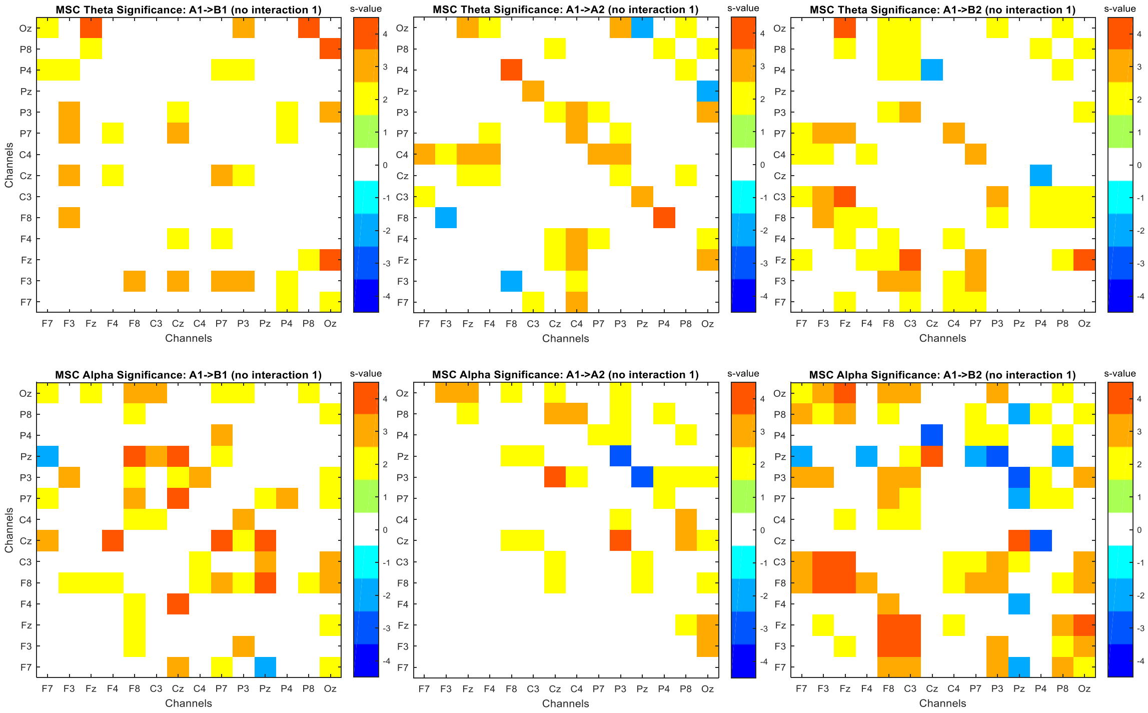
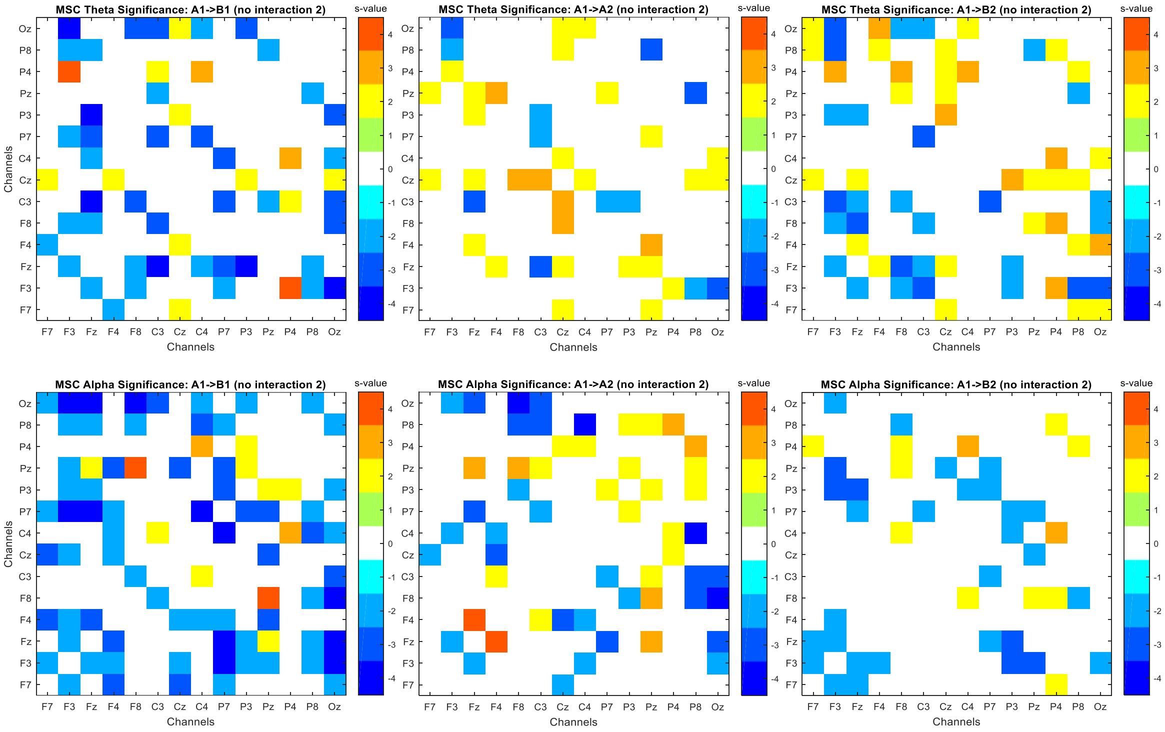


Figure 2-23. MSC significance in Subject 2 from the first baseline to each subsequent phase for the No Interaction 1 condition. MSC within the theta range is seen within the top 3 plots while MSC in the alpha range is seen in the bottom 3 plots.

Results from No Interaction 2 indicated that there was substantially more decrease in MSC than in No Interaction 1 for both frequency ranges. A1→ B1 for both ranges showed a decline in coherence between the parietal and frontal electrodes, as well as within the frontal lobe. A2 and B2 caused a mixture of increase and decrease in both theta and alpha. The MSC associated with the central midline rose during both phases in theta.


MSC Theta Significance: A1->B1 (no interaction 2)

MSC Theta Significance: A1->A2 (no interaction 2)

MSC Theta Significance: A1->B2 (no interaction 2)

MSC Alpha Significance: A1->B1 (no interaction 2)

MSC Alpha Significance: A1->A2 (no interaction 2)

MSC Alpha Significance: A1->B2 (no interaction 2)

Figure 2-24. MSC significance in Subject 2 from the first baseline to each subsequent phase for the No Interaction 2 condition. MSC within the theta range is seen within the top 3 plots while MSC in the alpha range is seen in the bottom 3 plots.

There was significantly more change in the intervention phases than during baseline. Both baseline phases for both frequency ranges show a scattering of significant change between intermittent pairs of electrodes. A2 in the alpha range shows a decrease between the central midline and the frontal electrodes. Both intervention phases in the theta range show reductions in MSC between the frontal and parietal electrodes and within the frontal lobe. B1 in the alpha range indicates decreased coherence between the parietal and central electrodes. B2 shows decreased correlation amongst all the frontal electrodes, as well as between frontal and left parietal electrodes

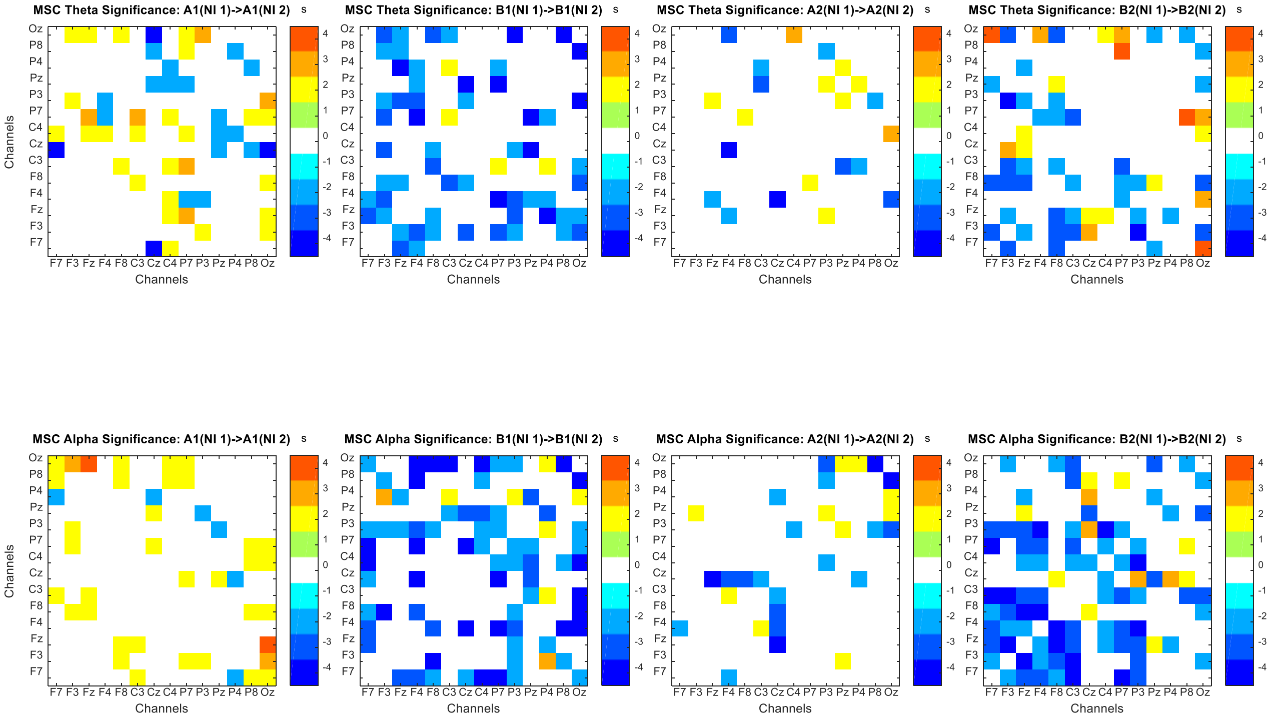


Figure 2-25. MSC significance in Subject 2 within each phase from the No Interaction 1 to No Interaction 2 condition. MSC within the theta range is seen within the top 4 plots while MSC in the alpha range is seen in the bottom 4 plots.

2.3.2.3 Mutual Information

Figure 2-26 shows significant changes in MI between electrodes for the No Interaction 1 condition. Overall, all 3 phases show a general elevation in MI. A1→ B1 show a cluster of increase between the parietal and frontal lobe. A2 appears to have caused intermittent increases in MI. B2 on the other hand shows a significant rise in MI for almost all electrode pairs.

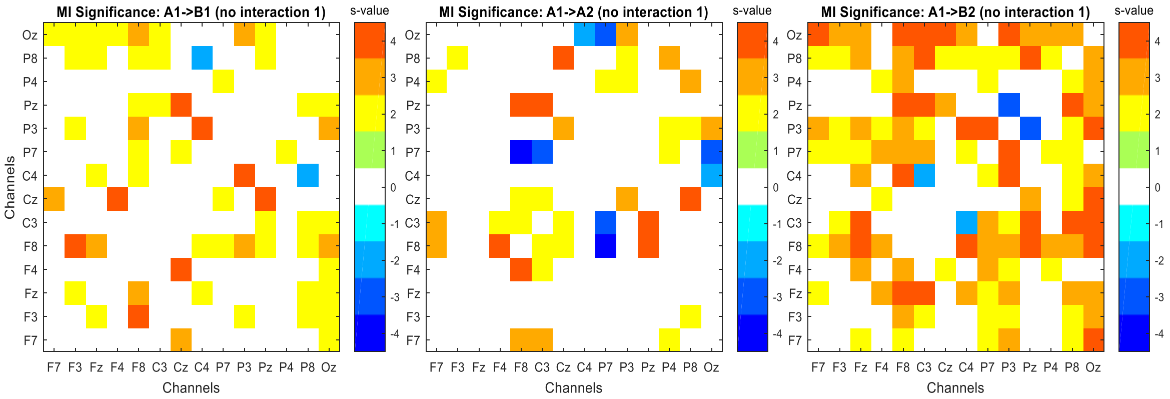


Figure 2-26. MI significance in Subject 2 from the first baseline to each subsequent phase for the No Interaction 1 condition

Figure 2-27 shows a decreasing trend in MI for B1 and an increasing trend for A2 during No Interaction 2. The number of electrode pairs that exhibited significant change was lower than that of No Interaction 1. However, A2 shows that there are small clusters of increased MI within the frontal lobe and within the parietal lobe.

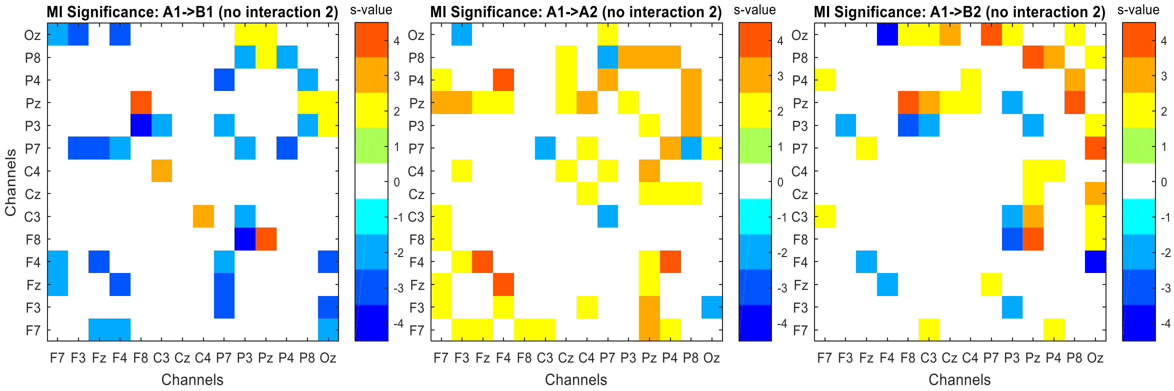


Figure 2-27. MI significance in Subject 2 from the first baseline to each subsequent phase for the No Interaction 2 condition

Lastly, MI was examined within each phase. Again, much more significant change was seen during the intervention phases than during baseline phases. B1 and B2 both show a significant reduction among many pairs of electrodes, namely within the frontal lobe (B2), within the parietal lobe (B1) and between the frontal and parietal electrodes (B1 & B2).

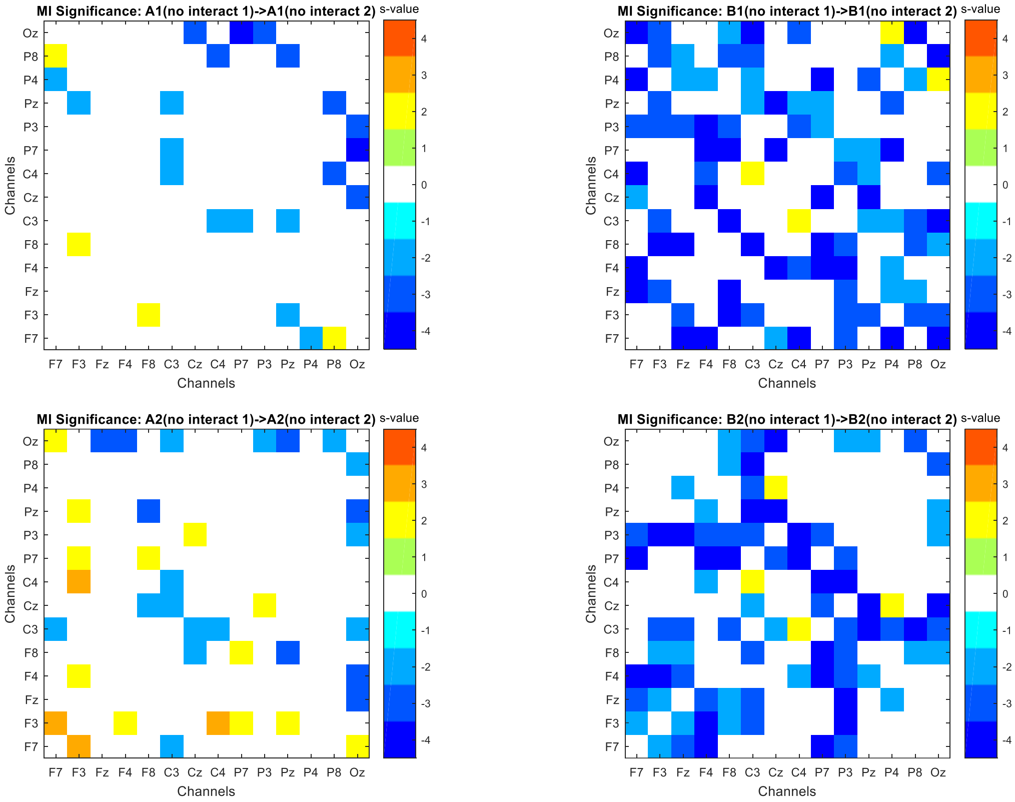


Figure 2-28. MI significance in Subject 2 within each phase from the No Interaction 1 to No Interaction 2 condition

2.3.2.4 Transfer Entropy

Figure 2-29 indicates that during the No Interaction 1 for A1→ B1 and A1→ B2, there was a rise in directed information transfer from the parietal electrodes to the central and frontal electrodes. Long term however, there was an increase in TE from the frontal lobe to the central and parietal lobe, as well as within the parietal lobe, and from the occipital midline to almost all other electrodes.

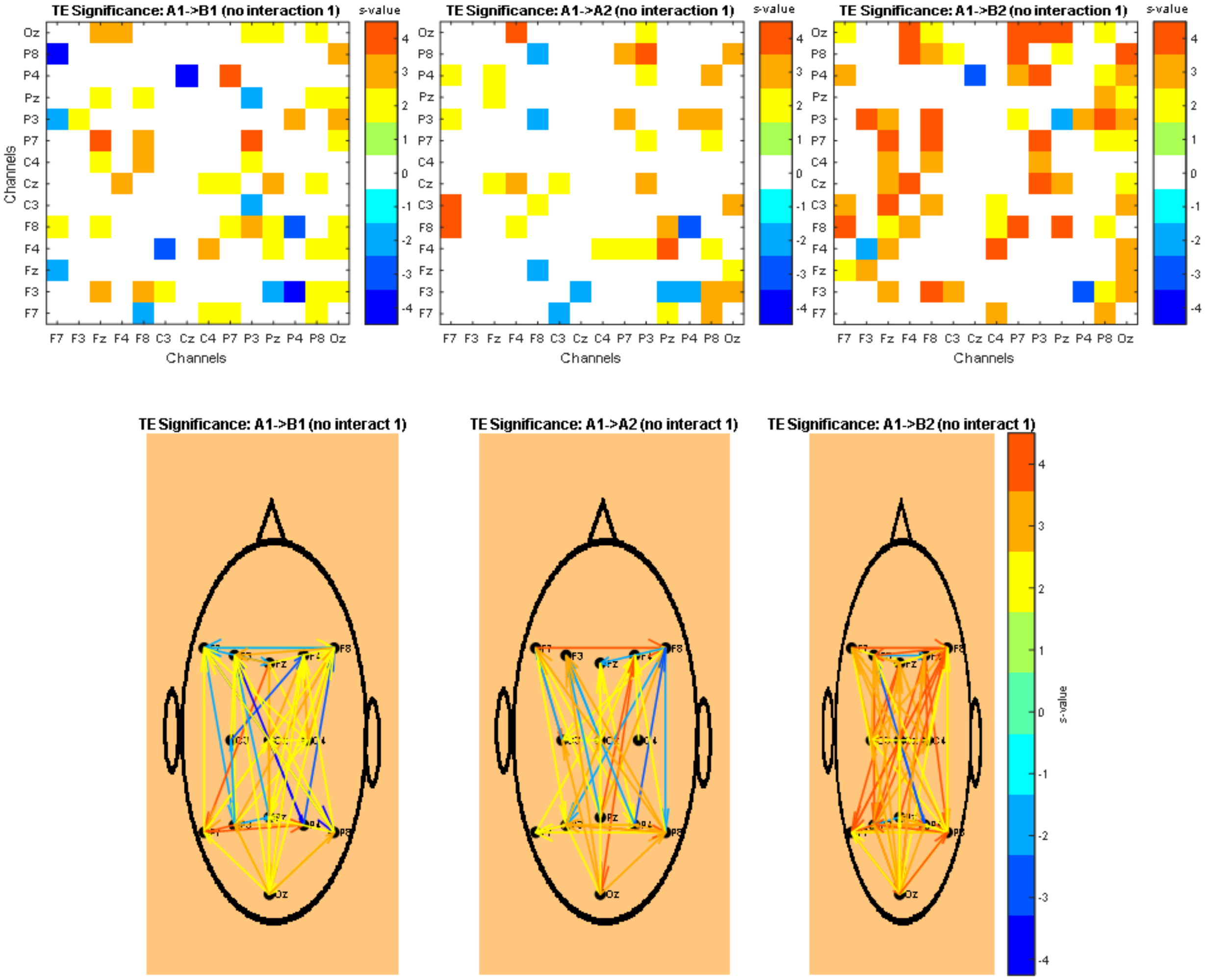


Figure 2-29. TE significance in Subject 2 from the first baseline to each subsequent phase for the No Interaction 1 condition. The top 3 plots depict TE in a heat map with the direction of information moving from *X*→*Y*. The bottom 3 plots also show TE using arrows from one electrode to the next.

TE analysis during No Interaction 2 revealed that there was little change in TE from A1→ B1 and A1→ B2, but substantial change from A1→ A2. TE rose from the frontal lobe to all other electrodes. There was also TE increase during B2 from ‘F7’ and ‘F4’ to the parietal and central electrodes.

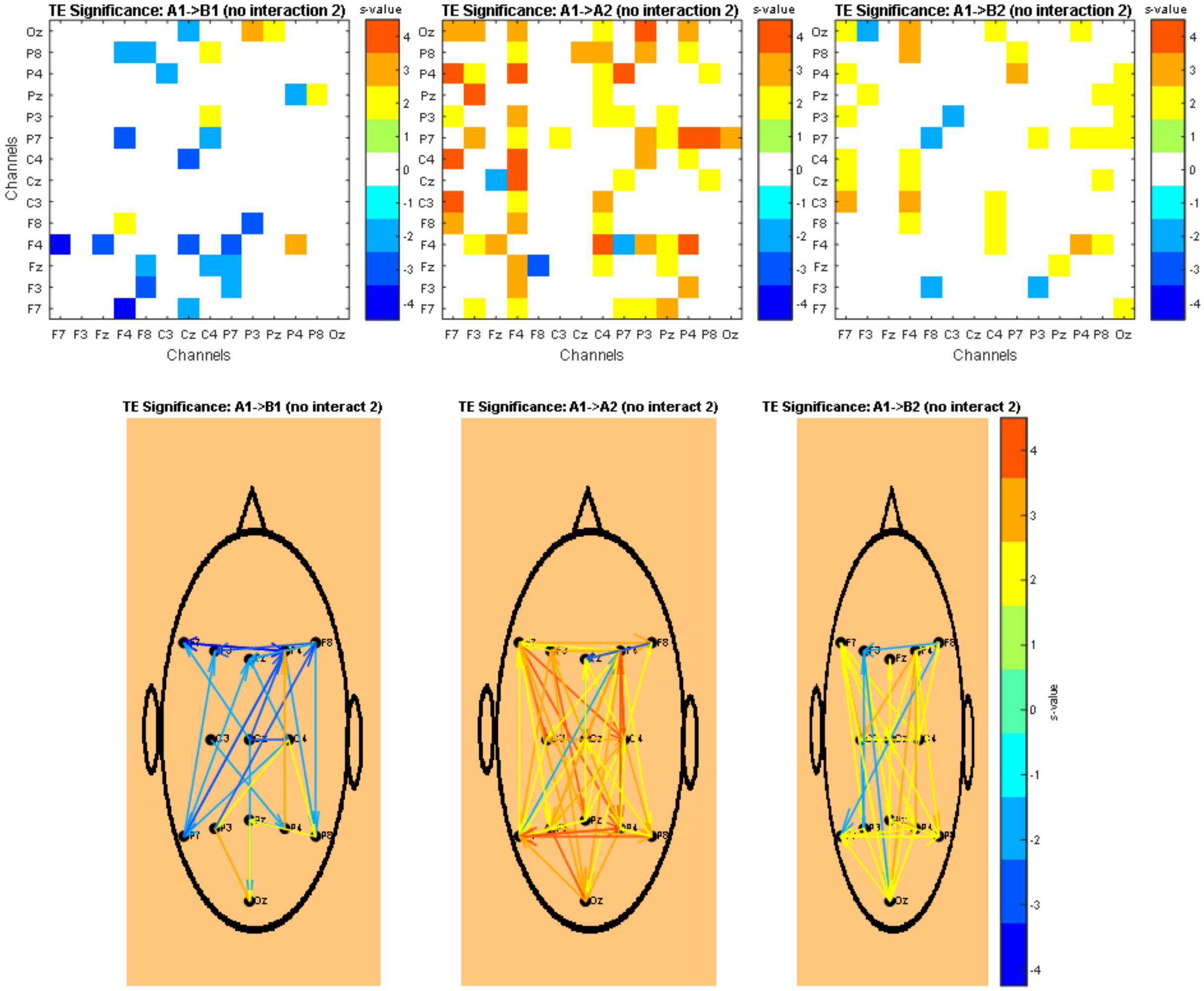


Figure 2-30. TE significance in Subject 2 from the first baseline to each subsequent phase for the No Interaction 2 condition. The top 3 plots depict TE in a heat map with the direction of information moving from *X*→*Y*. The bottom 3 plots also show TE using arrows from one electrode to the next.

Finally, TE was examined within each of the phases. Like the other metrics, baseline phases produced significantly less change than did intervention phases. TE increased from the frontal midline to the central and left parietal electrodes during A1 (top left). B1 (top right) in Figure 2-31 showed that there was a decline in TE from the frontal lobe to the central and parietal electrodes, and from the right central and left parietal electrodes to the other electrodes. B2 (bottom right) indicated that there was a reduction in information transfer from the right frontal to the left parietal electrodes, and from the left parietal to the central electrodes.

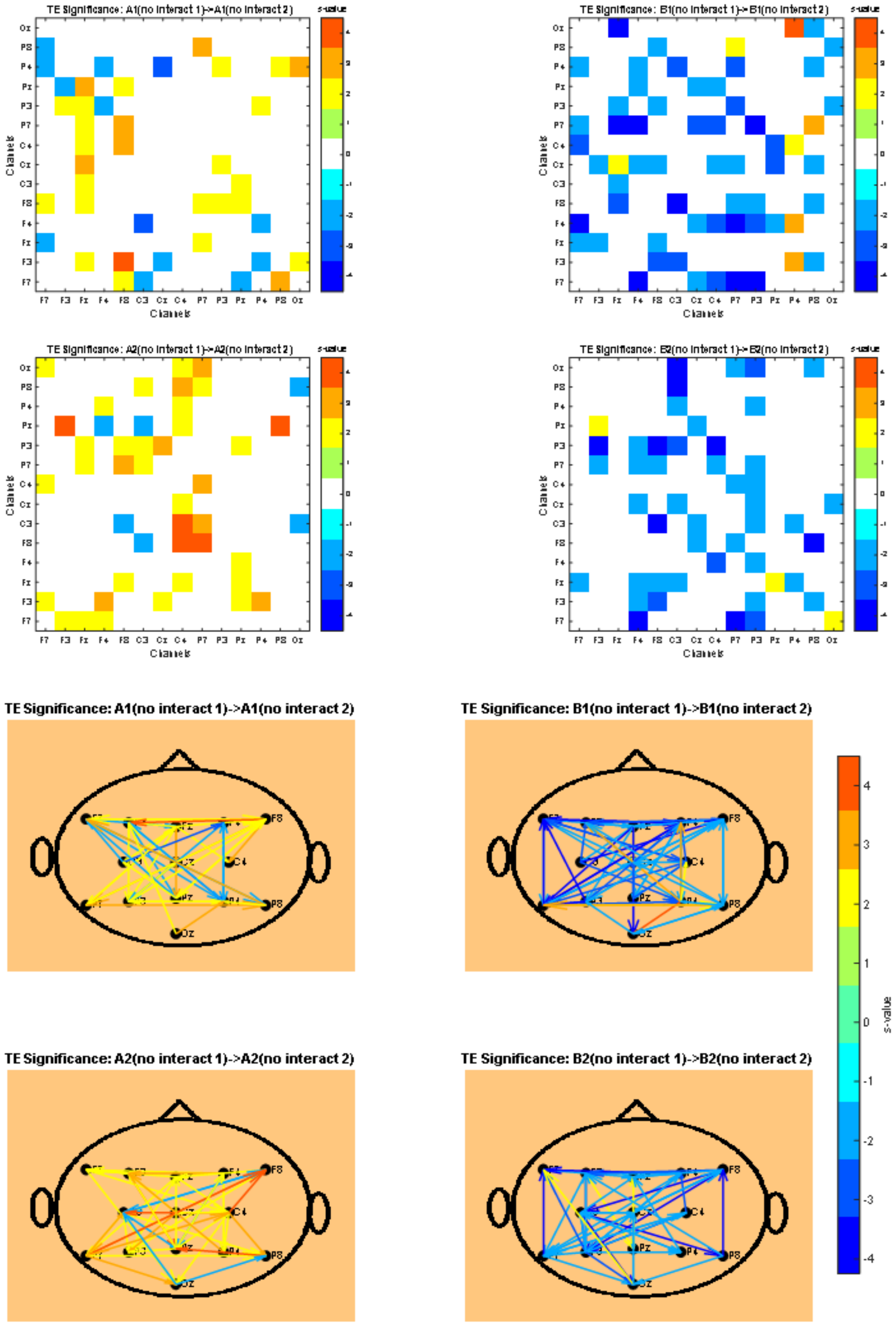


Figure 2-31. TE significance in Subject 2 within each phase from the No Interaction 1 to No Interaction 2 condition. The top 4 plots depict TE in a heat map with the direction of information moving from $X \rightarrow Y$. The bottom 4 plots also show TE using arrows from one electrode to the next.

2.3.3 Subject 3

2.3.3.1 Power Spectrum Density

Theta and alpha were identified based on the plots seen in Figure 2-32. Figures 2-32a and 2-32b illustrated that from No Interaction 1 to Interaction, increased spectral power was seen in the 2-5 Hz range. Additionally, the bottom two plots in Figure 2-32 indicate that there was a rise in PSD in that same range and a decrease in the frequencies above it. Therefore, the theta range was identified as 2-5 Hz and the alpha range was determined to be 6-11 Hz.

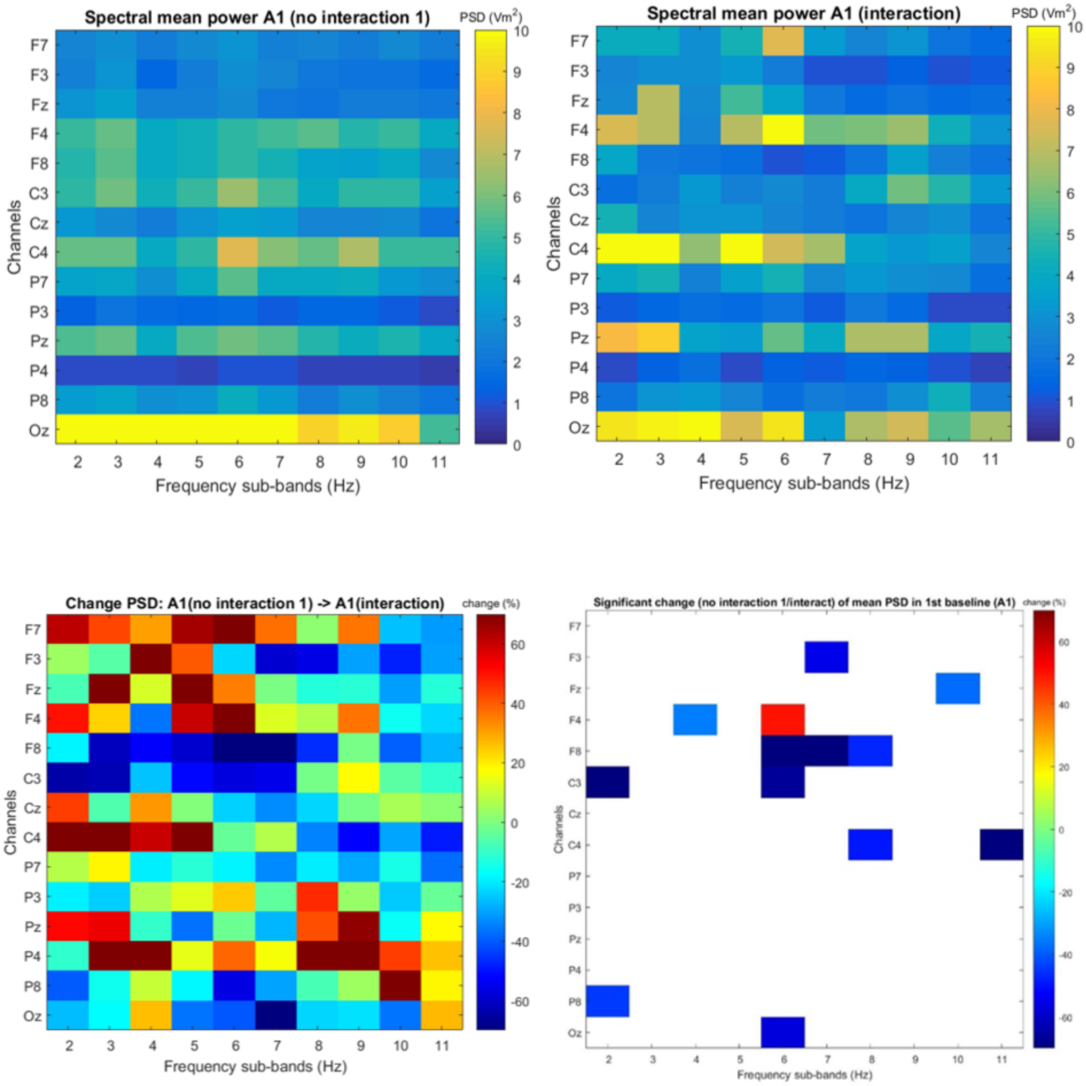


Figure 2-32 (a, b, c, d). Plots used to identify theta rhythm in Subject 3. Top left (a): spectral power during No Interaction 1. Top right (b): spectral power during Interaction. Bottom left (c): percent change in PSD. Bottom right (d): Significant percent change in PSD based on 2 standard error method.

Figure 2-33 demonstrates that for B1, theta power rose in the frontal electrodes while alpha power declined. Broadband increase in spectral power can be seen in ‘P4’ while broadband decrease is seen in ‘C3’ for both A2 and B2. B2 also shows a broadband elevation in spectral power in the left frontal electrodes and ‘P3’, as well as a reduction in alpha power in the central electrodes.

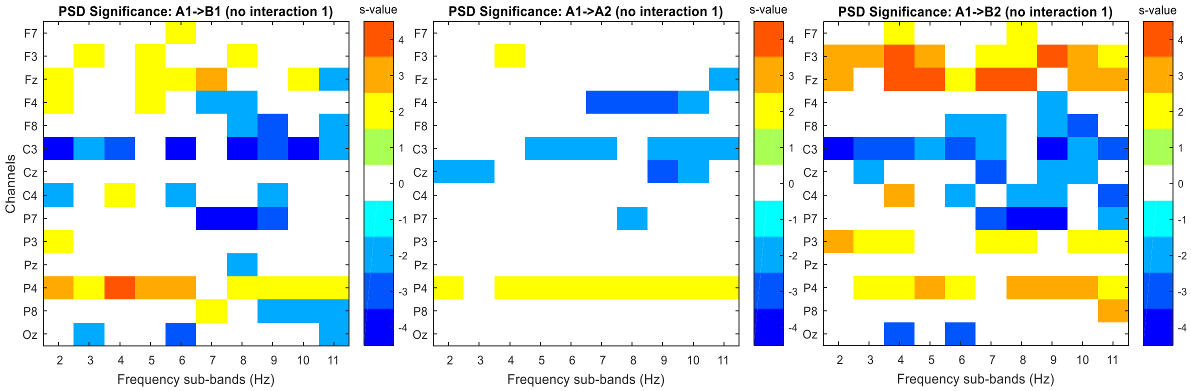


Figure 2-33. PSD significance in Subject 3 from the first baseline to each subsequent phase for the No Interaction 1 condition

No Interaction 2 produced more broadband activity. ‘F8’ experienced an increase in power from A1→ B1, and ‘Cz’, ‘P7’, and ‘P8’ experienced decreases as a result of phases B1 and B2 as well.

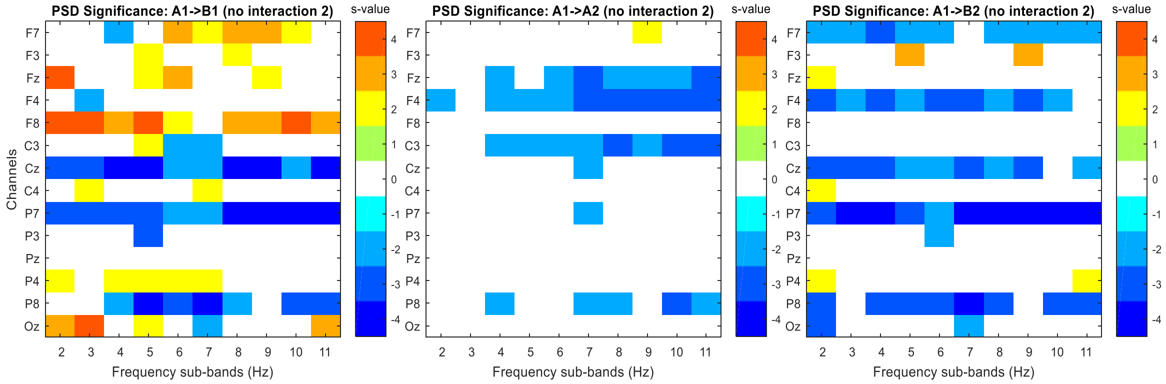


Figure 2-34. PSD significance in Subject 3 from the first baseline to each subsequent phase for the No Interaction 2 condition

The last condition examined for PSD was the change within each phase between No Interaction 1 and No Interaction 2. Figure 2-35 indicates that overall, more significant change occurred during intervention (B) phases than during baseline phases (A). Little change was exhibited during A1 (top left) except for broadband spectral elevation in the frontal midline (‘Fz’). Both intervention phases produced significant broadband spectral suppression in the parietal lobe and central midline. A reduction in power was also seen in the left frontal electrodes during B2. Spectral power rose in ‘C3’, ‘C4’, and ‘Oz’ for both intervention phases as well.

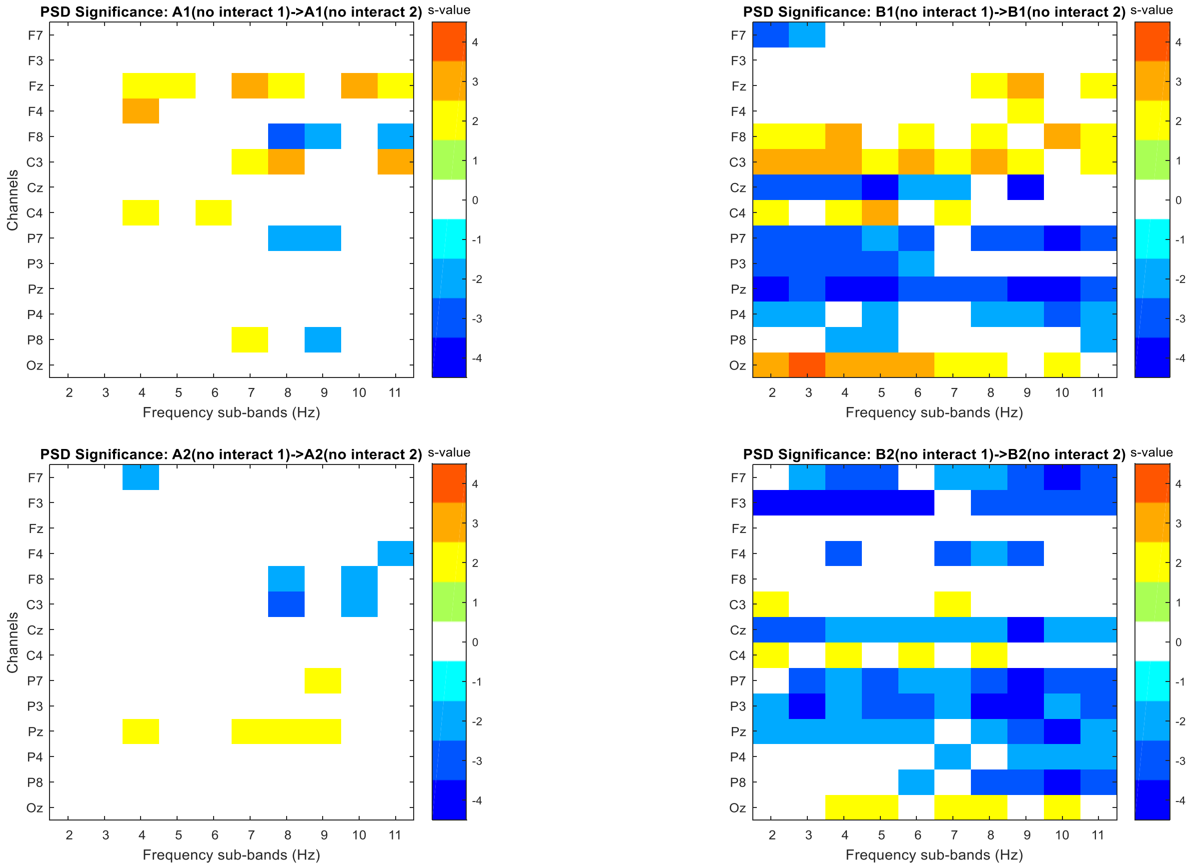


Figure 2-35. PSD significance in Subject 3 within each phase from the No Interaction 1 to No Interaction 2 condition

2.3.3.2 Magnitude-Squared Coherence

No consistent trends were seen during No Interaction 1 for the theta range. MSC appeared to decrease along the parietal midline as a result of B1. Long term, there were significant changes in MSC between the central and frontal electrodes. The alpha range saw a rise in MSC for all phases between the frontal and parietal lobes, as well as within the frontal lobe. B2 also demonstrated elevated coherence between the central and frontal lobes.

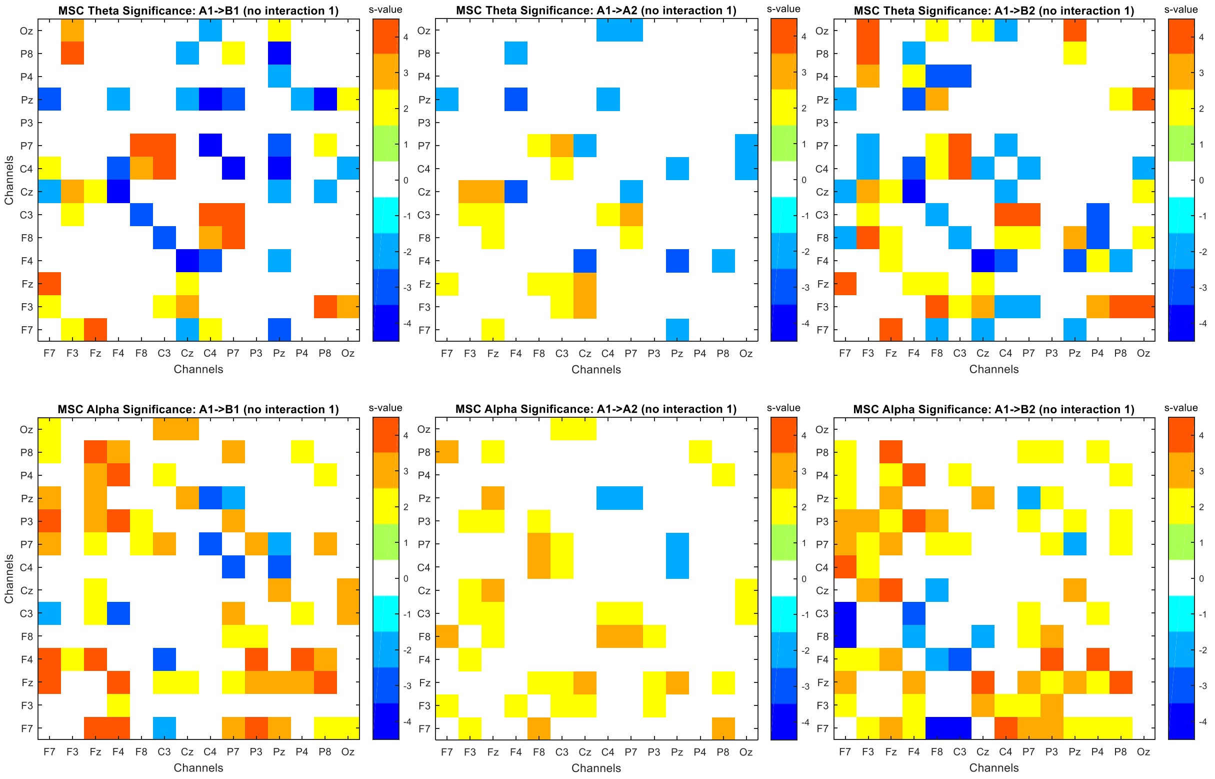


Figure 2-36. MSC significance in Subject 3 from the first baseline to each subsequent phase for the No Interaction 1 condition. MSC within the theta range is seen within the top 3 plots while MSC in the alpha range is seen in the bottom 3 plots.

No Interaction 2 essentially mimicked the results from No Interaction 1. There were no distinctive patterns for A1 and B1 in the theta range, but B2 showed an increase in MSC between the parietal and frontal electrodes, as well as within the parietal lobe. In the alpha range, MSC rose between the parietal and frontal electrodes from A1→ B1 and from A1→ B2, and additionally between the central and frontal electrodes in B2.

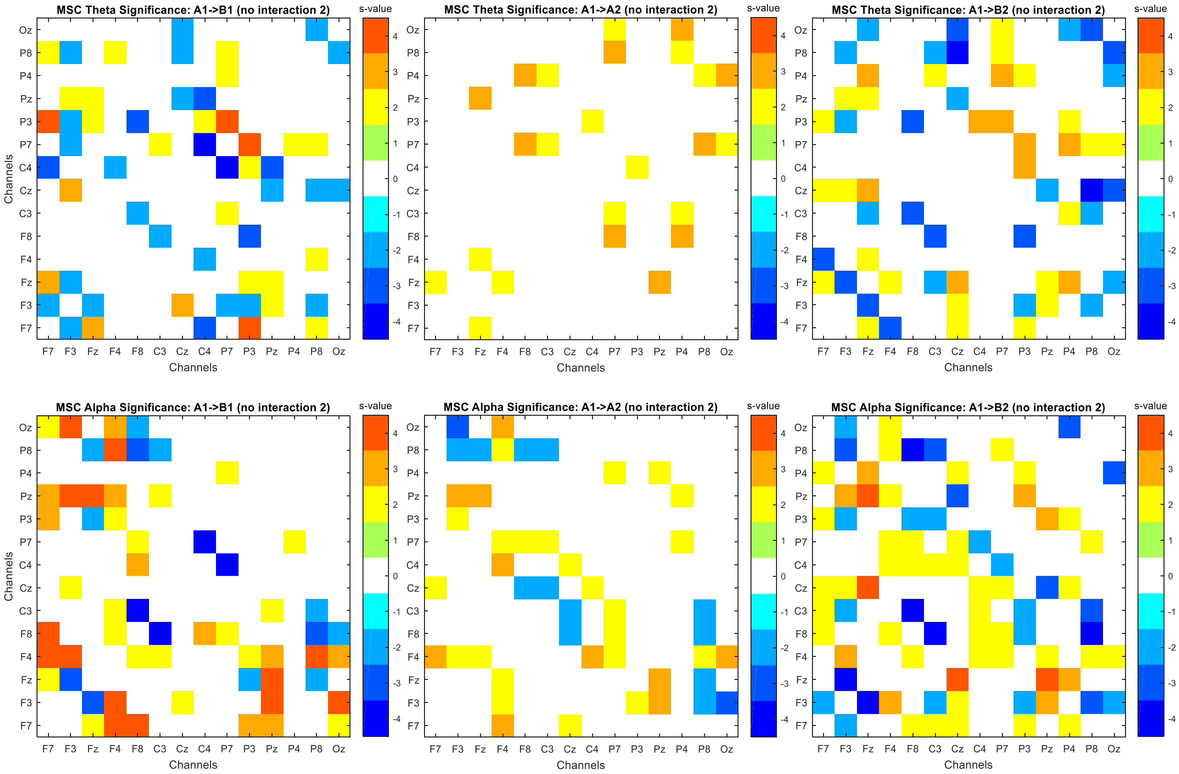


Figure 2-37. MSC significance in Subject 3 from the first baseline to each subsequent phase for the No Interaction 2 condition. MSC within the theta range is seen within the top 3 plots while MSC in the alpha range is seen in the bottom 3 plots.

MSC exhibited more change during the intervention phases than the baseline phases for both frequency ranges. The baseline phases indicate little change, but B1 for both phases show a reduction in coherence between the frontal and parietal electrodes. Figure 2-38 illustrates a reduction in MSC between the right parietal and right frontal electrodes, and the right parietal and left central electrodes in theta during B2.

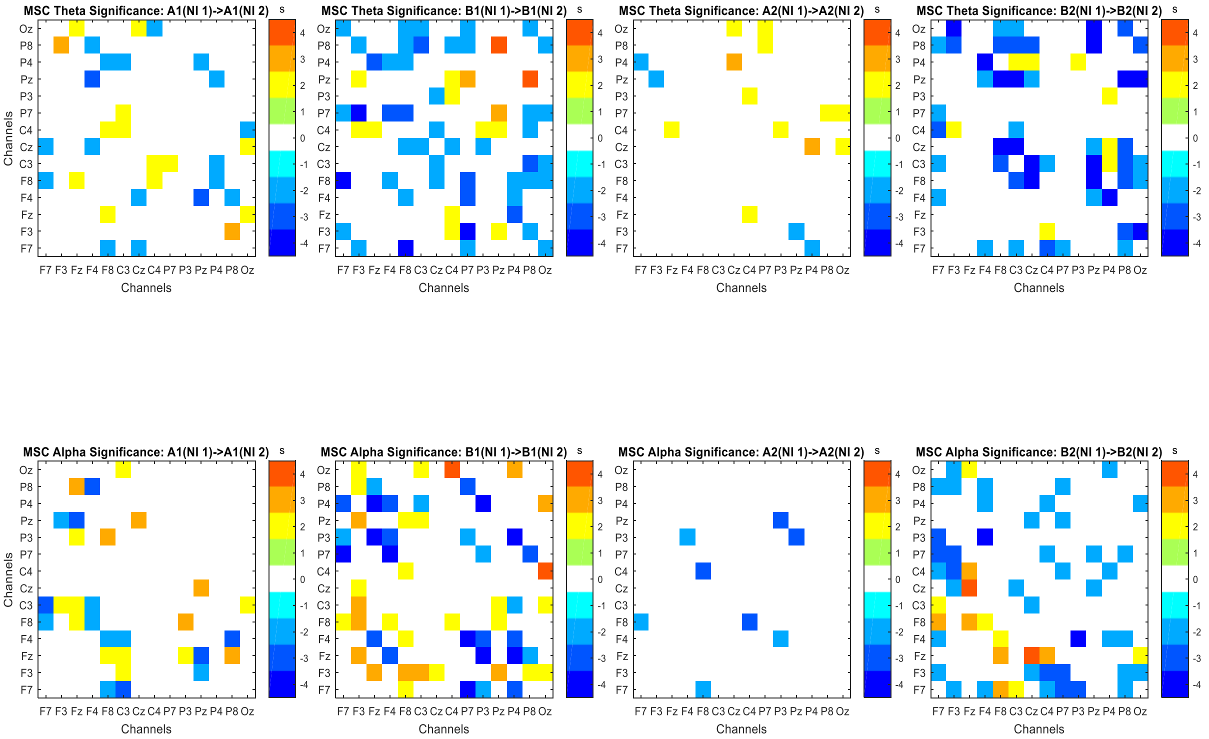


Figure 2-38. MSC significance in Subject 3 within each phase from the No Interaction 1 to No Interaction 2 condition. MSC within the theta range is seen within the top 4 plots while MSC in the alpha range is seen in the bottom 4 plots.

2.3.3.3 Mutual Information

Overall, all 3 phases from the original baseline showed an increase in MI. Moreover, both A2 and B2 show a rise in MI between the parietal and frontal electrodes.

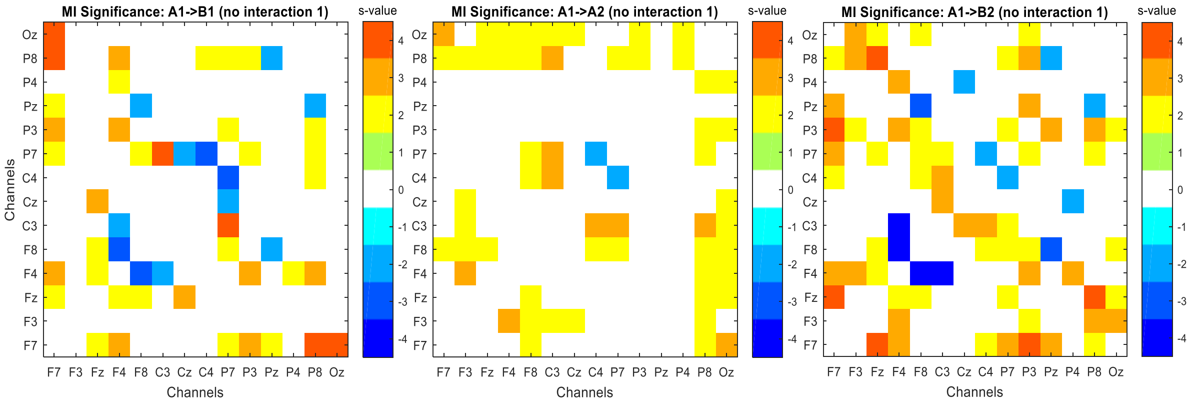


Figure 2-39. MI significance in Subject 3 from the first baseline to each subsequent phase for the No Interaction 1 condition

No Interaction 2 produced a general decline in MI between electrode pairs. The intervention phases indicate a reduction between the left parietal and right frontal electrodes, the left parietal and central electrodes, and within the frontal electrodes. A2 shows significant changes between intermittent pairs of electrodes.

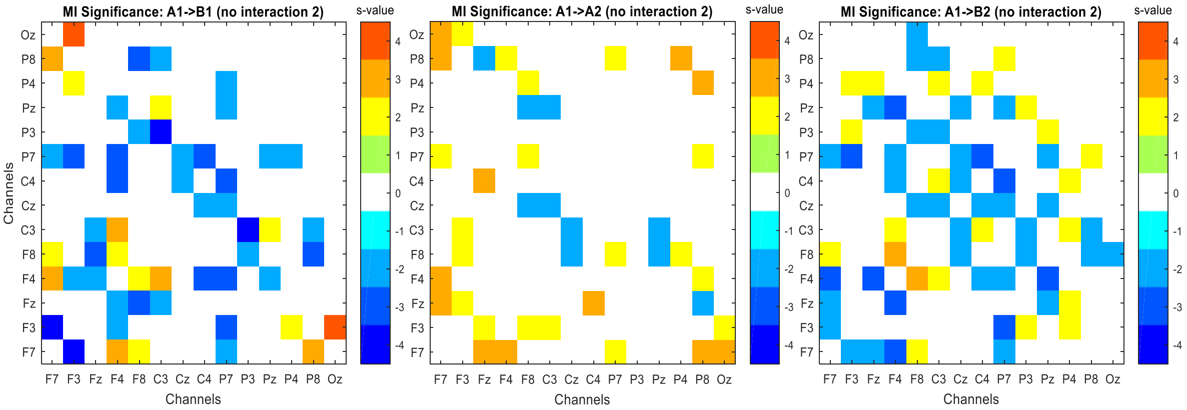


Figure 2-40. MI significance in Subject 3 from the first baseline to each subsequent phase for the No Interaction 2 condition

Lastly, MI was examined within each phase. Again, much more significant change was seen during the intervention phases than during baseline phases. B1 indicates changes between seemingly random pairs of electrodes, but B2 shows significant decreases in MI between the frontal and parietal electrodes.

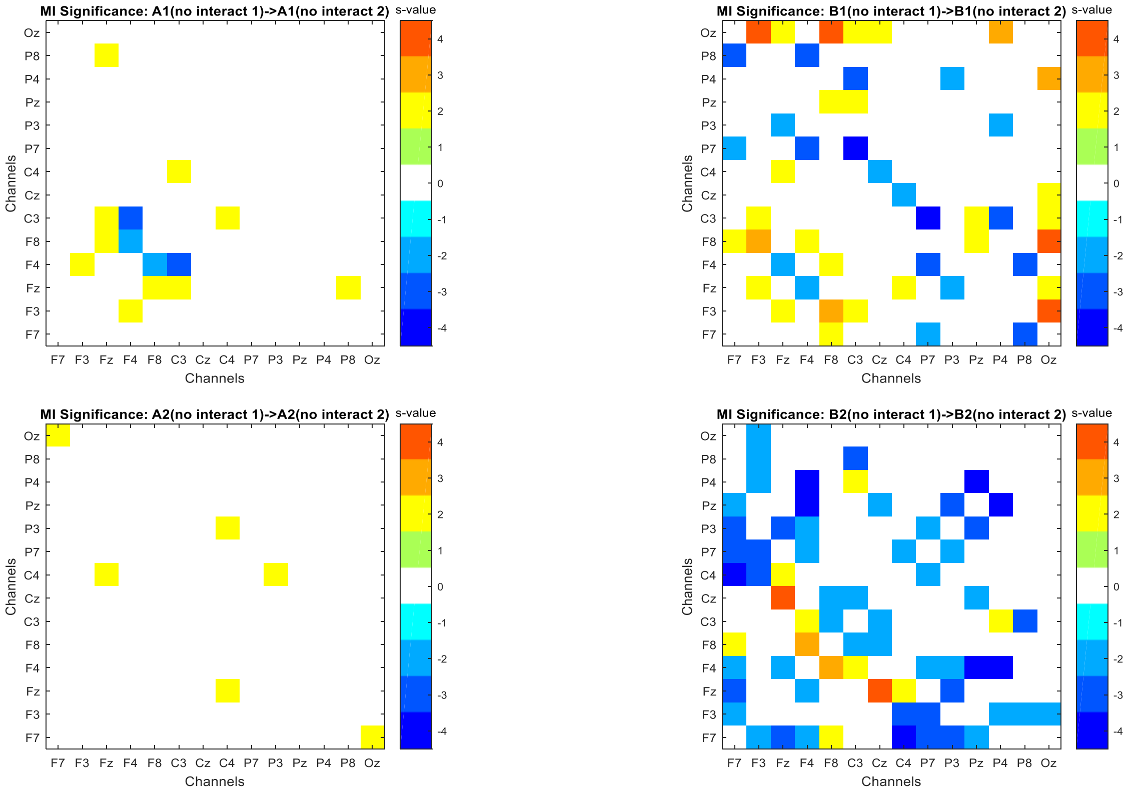


Figure 2-41. MI significance in Subject 3 within each phase from the No Interaction 1 to No Interaction 2 condition

2.3.3.4 Transfer Entropy

TE analysis from A1→ B1 during the No Interaction 1 condition exhibited a reduction in TE from the central electrodes to the parietal midline. All 3 phases after A1 indicate an increase in TE from the frontal to parietal electrodes, while A1→ B2 illustrated a rise in information transmission from the parietal electrodes to the frontal lobe as well.

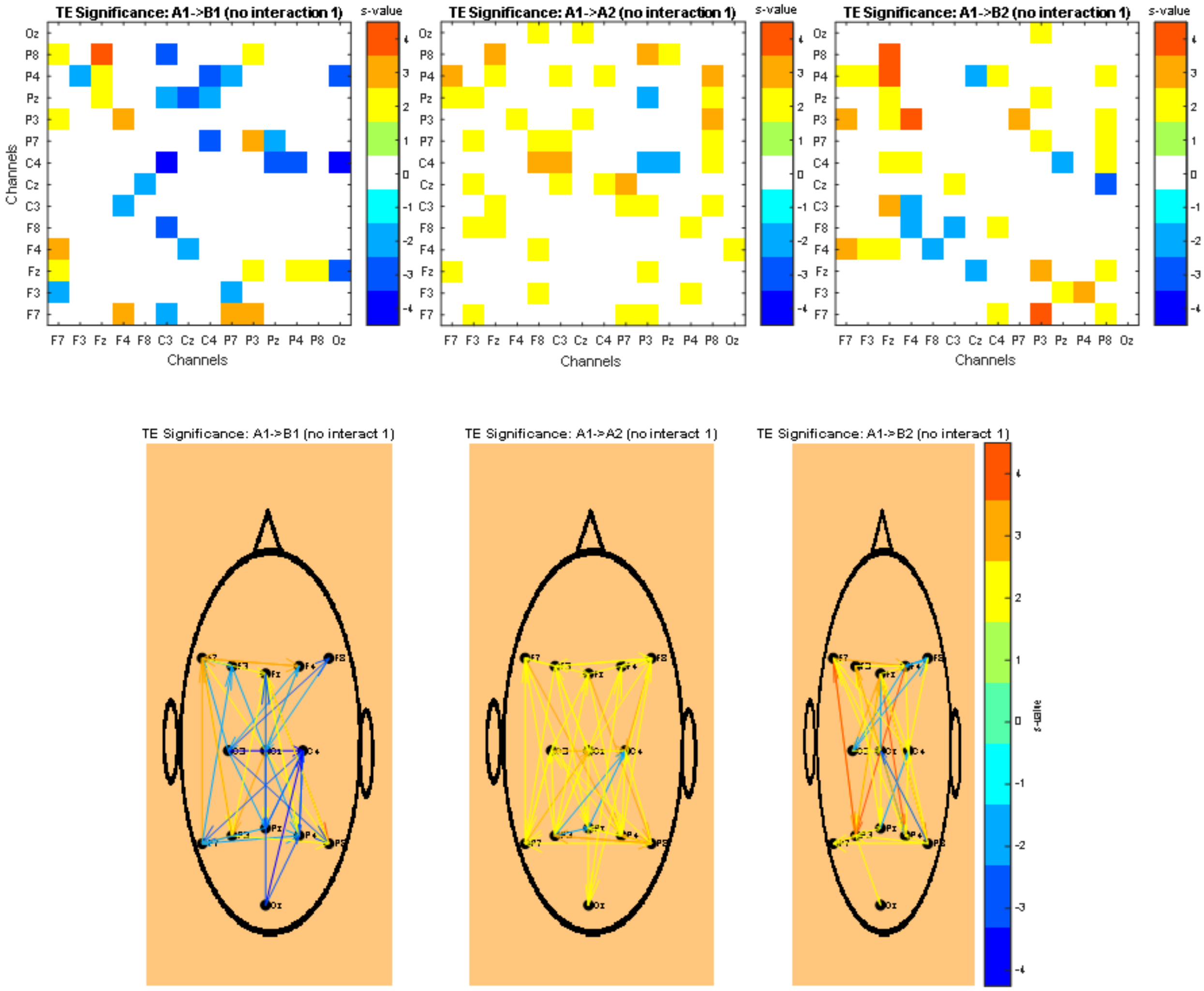


Figure 2-42. TE significance in Subject 3 from the first baseline to each subsequent phase for the No Interaction 1 condition. The top 3 plots depict TE in a heat map with the direction of information moving from *X*→*Y*. The bottom 3 plots also show TE using arrows from one electrode to the next.

TE analysis during No Interaction 2 revealed interesting trends in all 3 phases. A1→ B1 saw significant changes in TE from the occipital midline to the other electrodes. A1→ A2 produced a rise in TE from the central to frontal electrodes. A1→ B2 saw a reduction in TE from most electrodes to the central midline.

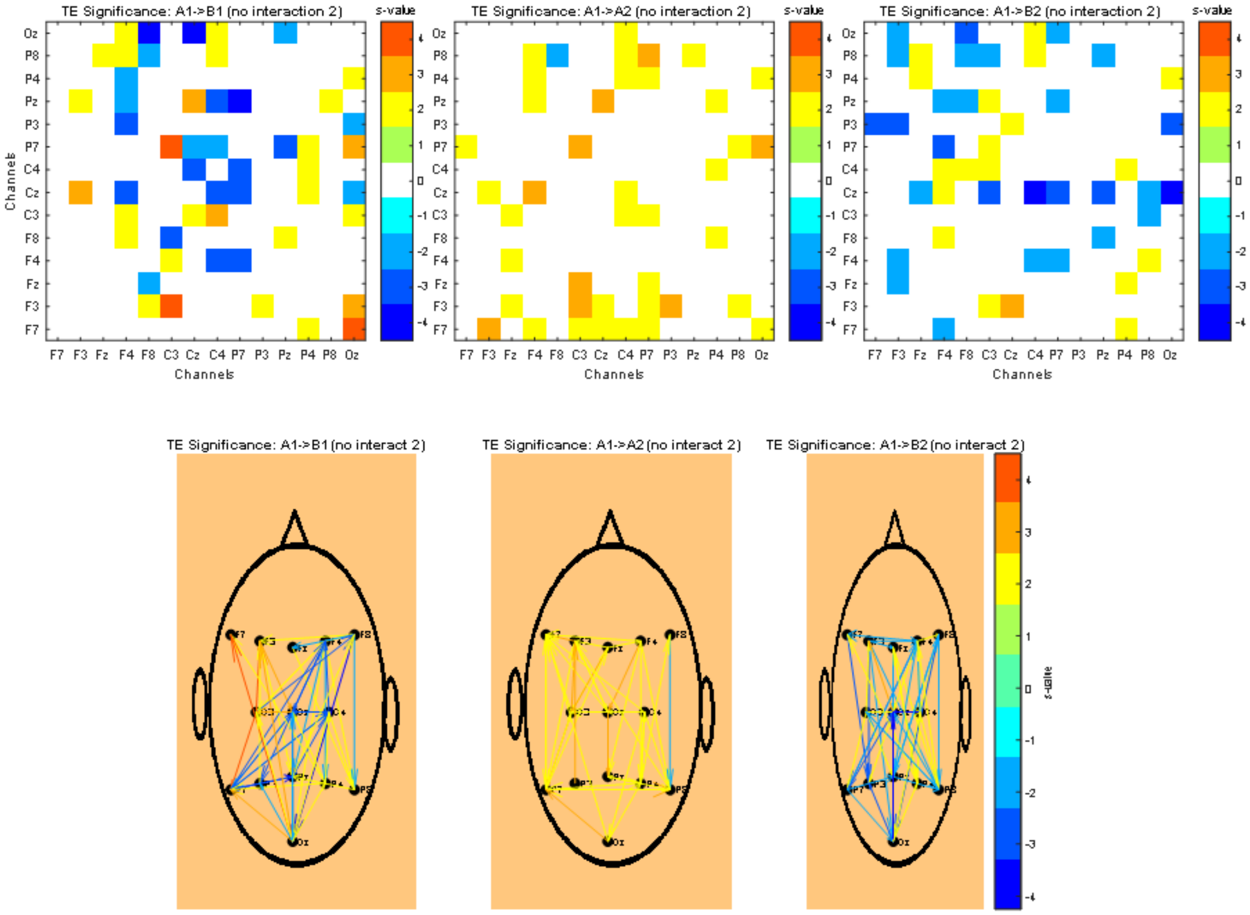


Figure 2-43. TE significance in Subject 3 from the first baseline to each subsequent phase for the No Interaction 2 condition. The top 3 plots depict TE in a heat map with the direction of information moving from $X \rightarrow Y$. The bottom 3 plots also show TE using arrows from one electrode to the next.

Finally, TE was examined within each of the phases. Baseline phases produced significantly less change than did intervention phases. TE did not show any trends during baseline, but decreased during intervention phases from the frontal to parietal lobes, vice versa, as well as between electrodes within each of those individual lobes.

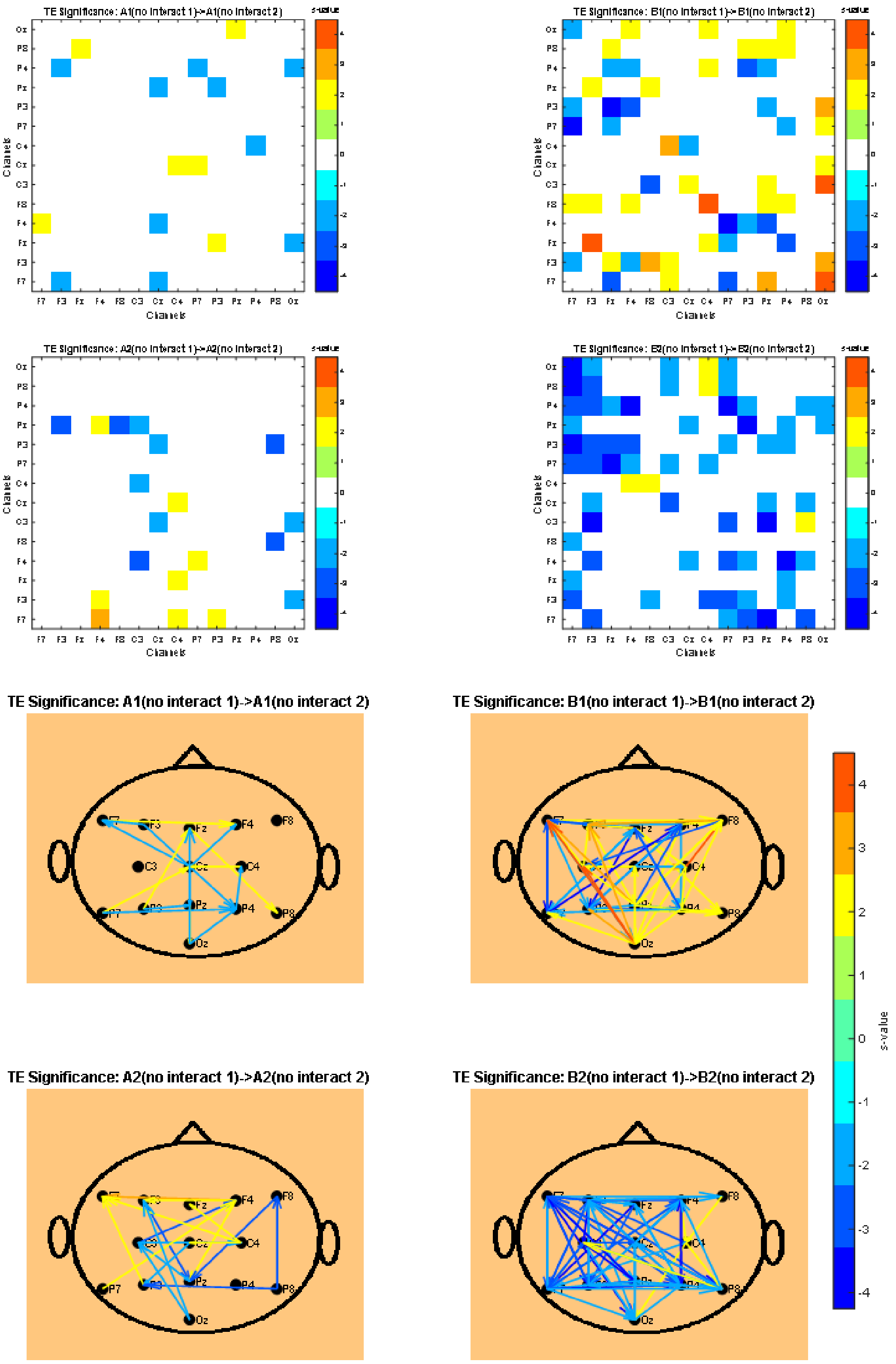


Figure 2-44. TE significance in Subject 3 within each phase from the No Interaction 1 to No Interaction 2 condition. The top 4 plots depict TE in a heat map with the direction of information moving from *X*→*Y*. The bottom 4 plots also show TE using arrows from one electrode to the next.

2.3.4 Rehabilitation Outcomes

Various metrics were observed from a rehabilitation standpoint and tracked over the course of the study. Table 3 displays each of the methods used to assess changes in each subject's physical capabilities.

Table 3. Rehabilitation results for each subject.

Participant	Access Method at Start of First Intervention (B) Phase	Access Method at End of Second Intervention (B) Phase	Wheelchair Skills Checklist Initial Final	ALP Phase Initial Final	COPM Performance Satisfaction
1	1 switch	2 switches	0 of 7 skills 4 of 7 skills	Phase 2 Phase 4	>2 points 5/5 >2 points 4/5
2	1 switch	2 switches	0 of 7 skills 4 of 7 skills	Phase 2 Phase 5	>2 points 5/5 >2 points 5/5
3	2 switches	3 switches	0 of 7 skills 4 of 7 skills & 1 emerging	Phase 3 Phase 5	>2 points 5/5 >2 points 5/5

Table 3 shows that all three subjects used switches to operate the Trainer and that each subject increased the number of switches used by 1 by the end of the study. Similarly, all subjects started off with a mastery level of 0 out of 7 skills in the Wheelchair Skills Checklist and progressed to at least 4 out of 7 skills over the course of the study. For the ALP, assessments were taken at the start and end of each of the intervention (B) stages. Both Subjects 2 and 3 started at Phase 2, 'Curious Novice', while Subject 3 started at Phase 3, 'Beginner'. Subject 1 progressed to Phase 4, 'Advanced Beginner', while both Subjects 2 and 3 elevated to Phase 5, 'Sophisticated Beginner'. Subjects 1 and 3 remained at the same phase from the end of B1 to the beginning of B2, while Subject 2 increased one phase in that same time frame. The COPM dictates that changes of 2 or more points in either performance or satisfaction is clinically significant. Table 3 shows that for all subjects, performance and satisfaction both increased significantly in at least 4 out of 5 of the activities that the parents deemed were important in power mobility development.

2.4 Discussion

In this study, changes occurring within the brain were investigated using four different metrics: PSD, MI, MSC, and TE. Based on previous findings, it was hypothesized that these metrics would be well-suited in quantifying changes in the level of cortical activity associated with cognitive and emotional states, as well as in associating these changes with rehabilitation outcomes. Several reoccurring themes were revealed while examining the results, but because this study made use of the SSRD method, the interpretation of the results have been partitioned by subject.

2.4.1 Subject 1

Most notable of the results was the level of involvement and interaction between the frontal and parietal lobes seen within each of the measures that were investigated. The frontoparietal network has been shown to be characterized by start-cue and error-related signals that suggests involvement in the control of initiation and adaptation during tasks [34,35]. Furthermore, studies have indicated that the frontoparietal network is engaged during repetition and memory recollection during reaching and visuomotor tasks in typically developing individuals [36,37]. During the intervention phases of the study, subjects underwent power mobility training where they would repeatedly reach for the switches that controlled the Trainer to navigate through hallways that included turns. This incorporated lots of starts and stops, as well as error correction if a turn was made incorrectly. In order to drive and navigate turns more successfully, the subject might need to recall their training, in addition to remembering their previous mistakes in order to avoid repeating them. Because these visuomotor tasks involved initiation and adaptation, as well as memory recall, it would be logical for the frontoparietal network to be heavily involved. Evidence of an active frontoparietal network can be seen in Subject 1 from the results of each of the four metrics. PSD analysis showed positive changes in theta power in the frontal and parietal lobes with concurrent negative changes in alpha power in the right hemispheric frontal lobe and left hemispheric central lobe. This occurred during the No Interaction 1 condition from baseline to all three phases which indicates that EEG rhythm changes were present in this network both long term and consistently over the course of the study. Coherence between the frontal and parietal lobes also rose in the theta range for No Interaction 2 and within the A2 phase, as well as in the alpha range for No Interaction 2. MSC also decreased within the frontal lobe in both theta and alpha ranges. It is suggested that an increase in coherence between the frontoparietal network in theta and upper alpha, as well as a decrease within the frontal lobe in alpha is reflective of an active network in functions of working memory [38]. Our results do indicate both complimentary and contradictory trends with previous findings however. While the MSC increase in theta within the network and decrease in the frontal lobe looks promising, other results are either inconsistent or contradictory. Additionally, the MSC was calculated as an average across each frequency range so it is therefore impossible to tell changes that occurred solely in the upper alpha frequencies. The combined results obtained from MI and TE analysis also point to a strong connection between the frontal and parietal lobes. Because MI is used to quantify the amount of shared information between two systems, and TE indicates the direction of information transfer, both metrics can be used to reveal functional connectivity within the brain [16]. For the No Interaction 1 condition, a long term increase in MI between the frontal and parietal lobes and within each of the two lobes was observed. TE from the frontal to the parietal lobe also rose. No Interaction 2 (except for A1→ A2) saw reductions for both metrics as did No

Interaction 1 → No Interaction 2 within the intervention phases. It is interesting to note the correlation between metrics being observed despite their underlying nature. PSD and MSC are both derived from frequency measures while MI and TE are purely statistically based, yet all four metrics showed similar trends in cortical activity within the brain. These findings suggest a significant interaction between the frontal and parietal lobes due to PMT and that information flow is directed from the frontal to parietal lobe. There are multiple frontoparietal networks just within theta however, but the data acquisition system used does not allow for the discernment of specific connectivity channels within a network [56]. Regardless, the emerging presence of a frontoparietal network may be a direct result of the visuospatial perception, motor skills, and memory recall/attention needs that power mobility training demands of children with severe disabilities.

Each stand-alone metric also produced interesting results. The simultaneous increase in theta and decrease in alpha power seen during No Interaction 1, which was also previously observed, may be associated with either decreased cognitive performance or increased anticipation and attention [1,39,40]. The latter is more plausible because the No Interaction 1 condition is at the beginning of the data collection day before any activities or tasks have been accomplished. The theta and alpha change could reflect the subject's excitement for the upcoming interaction and power mobility training that day. No Interaction 2 similarly produced minor theta increase and alpha suppression for A1→ B1 and A1→ A2 but also showed broadband spectral decrease in the parietal lobe from A1→ B2. The reason for this broadband decrease is unknown but literature suggests this may be due to ocular artifact [41]. In this study, it may also be caused by imbalances of the data acquisition system or daily condition of the subject [1]. No Interaction 1→ No Interaction 2 also produced broadband spectral decrease in the parietal lobe (B1 & B2) and left frontal lobe (B2). Again, this may be due to artifact, but it could also be due to a decreased anticipatory response before trials caused by the familiarity of power mobility training after 10 or 20 weeks of training.

MSC analysis produced few consistent trends. Positive change in coherence between the parietal and frontal lobes was observed in No Interaction 1 for alpha, which may be a result of attention-related amplification of neural responses [42]. As previously mentioned, this could be due to increased attention being paid at the beginning of each data collection session. The opposite could be said of No Interaction 2. Intermittent reductions of MSC were seen from A1→ B1 and from A1→ B2, as well as within A1, B1, and B2 from No Interaction 1→ No Interaction 2. The No Interaction 2 condition occurs at the end of each data collection session so it would make sense that less attention would be required by the subject, hence diminished inter/intra-lobe coherence. An increase in MSC from A1→ A2 for theta in No Interaction 2 was also perceived. Literature suggests that this phase coupling in theta across lobes with an associated increase in upper alpha could be reflective of the involvement of long-term memory [38,42]. Because upper alpha coherence could not be realized in this study however, it is impossible to attribute the concurrent changes in theta and alpha to any significant findings. In either case, phase coupling within the frontoparietal network may play an important role in central executive functions in working memory [38,42].

MI and TE are related quantities that detail how much information one signal contains about another (MI) and the direction in which information is transferred between signals (TE). In both metrics, we see the familiar correlation between the frontal and parietal lobes. In all phases for No Interaction 1, MI and TE both increased, with TE depicting that information was flowing from the frontal to the parietal lobe. No Interaction 2 saw a general reduction in MI across electrode pairs from A1→ B1 and A1→ B2 and within the intervention phases from No Interaction 1 → No Interaction 2. TE saw a similar trend and depicted the decrease in information transfer as occurring from the frontal to parietal lobe. The frontal lobe is known to be involved in a variety of cognitive processes including executive function, attention, memory, and motor function, and is vital in carrying out higher order processing [43,44]. The parietal lobe on the other hand, plays a large role in visuomotor tasks including the understanding of intention, spatial attention, and episodic memory retrieval [45-47]. These functions in the parietal and frontal lobes are heavily utilized during power mobility training so the observance of changes in shared information and information transfer between the two lobes is not surprising. The direction of information flow from the frontal to parietal lobe might be due to the frontal lobe's role as a processing center that assesses the current situational needs and disseminates information and commands out to the necessary regions, i.e. the parietal lobe. The increase in MI and TE during the No Interaction 1 condition may be explained by similar trends seen in the other metrics. The subject is not performing any task during No Interaction 1, but perhaps because of the anticipatory response to starting power mobility training, more cortical resources are being recruited in preparation for the tasks ahead. Memory recall from previous training sessions might be occurring, as well as increased attention. Conversely for No Interaction 2 and from No Interaction 1→ No Interaction 2, the training has been concluded for that session, causing a decrease in attention and a cease in visuomotor skill use, negating the need to recruit additional cortical resources. Information transfer is almost certainly still occurring, but at a significantly lower level than before. This could also help explain the long-term decreases seen. Towards the end of the study, the subject may have become more familiar with training procedures and therefore did not need as many cognitive resources to complete the same tasks.

This may also explain why much more significant change is seen within the intervention phases as opposed to the baseline phases in all four metrics. Because the baseline phases do not include power mobility training, it is likely that power mobility training is responsible for an overall change in cortical activity. It was also interesting to note the significant changes in TE from the occipital midline to other electrodes. Kakizaki theorized that an increase in occipital midline beta-2 amplitude in EEG may be modulated as a function of task difficulty [48], but because spectral analysis only focused on the theta and alpha ranges, this finding could not be corroborated with the trend observed in TE.

Examination of the four rehabilitation outcomes revealed significant improvement in Subject 1's ability to control the Trainer. Subject 1 was able to navigate an additional direction and could successfully perform more wheelchair skills. While the access method, Wheelchair Skills Checklist, and COPM all observe a child's ability to perform specific tasks or obtain certain skills, the ALP gives more insight into a broader view of a child's level of understanding and learning, as well as emotional and social development. The initial ALP categorized Subject 1 as residing in Phase 2 ('curious novice'), in which the child shows occasional alertness, is responsive to social interaction, and is beginning to explore the concept that a self-initiated act can cause motion in the Trainer [61]. By the end of the study, Subject 1 had progressed to Phase 4 ('advanced beginner'), defined by the child being attentive, requesting interaction (by pointing at people and objects), and understanding that 2 switches have different functions and create motions in different directions. This increase in attentiveness and retention in learning correlates to changes that were seen in the EEG measures. The increase in theta power and suppression in alpha power, which has been linked to increased alertness, is reinforced by the rehabilitation data. Additionally, the fact that Subject 1 stayed at the same phase during the second baseline

(A2) indicates that previous learning and understanding had been retained. This could help explain why the EEG metrics still experienced an increase from A1→A2 even though no Trainer was used. The proposition of an increased frontoparietal network presence is strongly supported by the development in understanding and learning of the Trainer indicated by the ALP. Because of the frontoparietal network's involvement in visuomotor tasks, it is logical that an enhanced understanding of Trainer use would correspond to a more active network, reflecting the increased recruitment of cortical resources to better manipulate the power wheelchair.

2.4.2 Subject 2

Subject 2 exhibited trends that were similar to Subject 1. The emergence of a frontoparietal network was evident in MI, MSC, and TE. MSC increased between the frontal and parietal lobes in alpha for No Interaction 1 and decreased in alpha for No Interaction 2. Again, this could be due to a rise in attention and anticipation for No Interaction 1 at the advent of power mobility training. MI and TE also captured frontoparietal activity through an increase in both metrics for No Interaction 1, with TE depicting information transfer from the frontal to parietal lobe (A1→ B2). Like Subject 1, there was also a reduction in MI and TE within the intervention phases from No Interaction 1 to No Interaction 2. Like Subject 1, this decrease might be a result of less cortical stimulation from the beginning of training that session to the end. Additionally, the subject could have been in a more subdued state as a result of fatigue from the training session. While the frontoparietal network appeared to have been present and active, it seems interactions between the two lobes were not as consistent or prominent as they were in Subject 1. What was equally as prominent in Subject 2 however, was an active centroparietal network. PSD analysis revealed significant changes in both the central and parietal regions throughout No Interaction 1, No Interaction 2, and within the B2 phase from No Interaction 1→ No Interaction 2. MSC also supported this claim with increased phase coupling between the central and parietal regions in alpha in No Interaction 1 and decreased coherence in alpha for No Interaction 2. Studies suggest that phase modulation in EEG rhythms, particularly alpha, as well as enhanced centroparietal coupling in slow and fast rhythms are associated with anticipation in somatosensory and visuomotor events [57]. MI and TE also indicated shared information and information transfer between the two regions. Both metrics showed increases in No Interaction 1, with TE indicating that the flow of information traveled from the parietal lobe to the central region from A1→ B1. Within the intervention phases, MI and TE both decreased from No Interaction 1 → No Interaction 2 as seen in the other metrics. These findings point to the presence of a centroparietal network in which proprioceptive information is integrated in the parietal lobe and sent to the central motor system to transform that information into operative motor commands [57].

Analysis of individual metrics also produced promising results. In addition to the spectral changes seen in the central and parietal regions, No Interaction 1 saw a suppression of theta power in the central midline and increase in alpha power in the right central region as well as in the parietal lobe from A1→ B1. There was a similar pattern in No Interaction 2 from A1→B1 and from A1→ B2 where decreases in theta power and increases in alpha power were seen in the centroparietal network. The B2 phase from No Interaction 1→ No Interaction 2 also saw this pattern. This oscillating theta and alpha pattern has been observed in previous studies and attributed to an increase in cognitive performance [40]. These findings could therefore suggest that there was a short-term increase in the subject’s cognitive functioning as a result of the B1 phase for No Interaction 1 and No Interaction 2, as well as the B2 phase for No Interaction 2 and within the power mobility training during B2. Because this trend wasn’t observed during baseline phases, one could infer that this increase in cognitive performance was a direct result of power mobility training. PSD changes in the remaining conditions either showed broadband changes in spectral power likely due to ocular artifact, or saw an increase in theta and a simultaneous decrease in alpha power. As previously mentioned, this could either be associated with decreased cognitive performance, or more likely, an increase in anticipation and attention [39,40].

MSC analysis revealed an increase in phase synchrony between central and frontal regions in theta for No Interaction 1, as well as between the centroparietal network and frontoparietal network in the alpha range. Again, the change in upper alpha could not be isolated in order to infer the involvement of long-term memory during power mobility training, but increases in theta and alpha individually may be related to higher attentiveness [42]. MSC was reduced in No Interaction 2 in the frontoparietal network which might reflect a decrease in attention towards the end of training days. Anticipation and attention in No Interaction 2 during the first baseline could still have been at high levels because the training environment was still novel to the subject. The decrease in coherence seen in subsequent phases would likely be caused by increased familiarity with the study, evoking less anticipation and requiring less attention.

Information theoretic results mirrored that of the spectral findings. MI and TE both increased in the frontoparietal (B2) and centroparietal (B1) networks for No Interaction 1. TE indicated that information flowed from the parietal lobe to both the central and frontal regions from A1→ B1, and from the frontal lobe to both the central and left parietal regions from A1→ B2. This result may suggest an active centroparietal and frontocentral network due to increased anticipation regarding a sensorimotor event (centroparietal) as well as the involvement of motor memory (frontocentral) [57,58]. TE also increased from the frontal lobe to most other electrodes from

A1→ A2 for No Interaction 2, which may support the notion that the frontal lobe is vital in processing information from various regions of the brain and carrying out executive functions during visuomotor tasks [43,44]. Similar to Subject 1, the intervention phases saw a reduction from No Interaction 1→ No Interaction 2. MI decreased in in the frontoparietal and centroparietal networks, as well as within the individual frontal and parietal lobes. TE saw less directed information transfer from the frontal, parietal, and central electrodes to others. This decrease within the networks could again be due to the subject starting off training at a high anticipatory and attentiveness level, necessitating increased cortical activity, to a more subdued state after the training was complete. Because much more activity was seen within intervention phases than during baseline, one could infer that significant changes in MI and TE, as well as in PSD and MSC, were due to power mobility training.

Subject 2 also demonstrated improvement in all rehabilitation measures, especially the ALP. Initial assessment placed Subject 2 in Phase 2, while the final assessment placed the subject in Phase 5 ('sophisticated beginner'), which is characterized by active concentration, triadic or reciprocated interaction, and deeper exploration and experimentation with the capabilities of the Trainer [61]. Like Subject 1, this also provides another explanation for the emergence of a frontoparietal network. Subject 2 demonstrated the ability to apply retained learning by experimenting with the steering of the Trainer, which could suggest a correlation with the increased activity in centroparietal network. Experimental and more intentional steering require a heightened sense of proprioception to understand where the body is spatially positioned. In power mobility training, having this increased perception of where oneself is would allow for more focused and directed motor tasks to facilitate desired movement. The modulation of this feedback loop has been shown to be associated with the centroparietal network [57]. Subject 2 also progressed by a phase from the end of B1 to the start of B2, suggesting that although there was no physical use of the Trainer, Subject 2 was still actively processing what had been learned. The suppression of theta and increase in alpha power, a trend commonly associated with increased cognitive performance, that was observed in Subject 2 over the course of the study may supplement the rehabilitation findings of a persistent growth in learning and understanding.

2.4.3 Subject 3

Subject 3 exhibited trends more similar to Subject 1 than Subject 2. There was a strong indication of the presence of an active frontoparietal network. PSD analysis did show broadband spectral increases in the frontal and parietal lobes, along with broadband suppression in the central region for No Interaction 1, but these results may have artefactual origins [41]. There was a strong correlation between the spectral phases in the parietal and frontal lobes within the alpha frequency range for both No Interaction 1 and No Interaction 2. This was mainly exhibited for A1→ B1 and A1→ B2. Furthermore, MSC decreased in the frontoparietal network within the B1 phase for No Interaction 1→ No Interaction 2, likely signifying the change in anticipation and attention from the beginning of the training session to the end [36,37]. MI and TE both showed long term increases across No Interaction 1 and reductions within the B2 phase between the two lobes, with TE depicting the flow of information transfer from the frontal to parietal lobe. These two metrics reveal that power mobility training may cause significant changes in communication to occur within this network due to its association with visuomotor tasks, attention, and memory [34-38].

Individual metrics in Subject 3 revealed trends similar to those seen in the other two subjects. PSD interpretation proved futile due to the presence of broadband spectral changes. There wasn’t a strong indication of theta and alpha oscillations that would suggest changes in cognitive performance or attention. MSC did reveal positive changes in synchrony between parietal and frontal lobes, as well as between the central and frontal regions in alpha which is likely a result of increased attention and anticipation. The decreases seen within the intervention phases between the parietal and frontal lobes (B1) and parietal and central regions (B2) may also be due to a decrease in anticipation and alertness over the course of the training session and study caused by fatigue and familiarity as previously suggested. MI and TE revealed trends that were consistent in this study in that there was a change in shared and directed information between the frontal and parietal lobes. TE additionally indicated a reciprocal decrease in information flow within B2 from the frontal to parietal lobes and vice versa, as well as within the parietal lobe. All 4 metrics displayed significantly more change as a result of the intervention phases, leading to the conclusion that power mobility training may be directly involved in these changes seen in cortical activity.

Subject 3 also demonstrated progress in all the rehabilitation metrics, including an elevation from Phase 3 to Phase 5 in the ALP. While the EEG measures did not change consistently in the short-term, the long-term results that indicated the presence of an active frontoparietal network correlated well with the observed developments in learning and understanding through power mobility training.

In summary:

- An active frontoparietal network was revealed in all three subjects, signifying the recruitment of cortical resources for visuomotor function, attention, and memory during power mobility training [34-38].

- An active centroparietal and frontocentral network may also have been seen in Subject 2, suggesting involvement of anticipation and memory related to sensorimotor events [57,58].
- There was significantly more change seen during the intervention phases in all 4 metrics compared to the baseline phases in all 3 subjects, suggesting that power mobility training was at least somewhat responsible for the differences observed in cortical activity.
- There was a general trend of increase in MSC and information transfer seen across No Interaction 1 and decrease across No Interaction 2 over the course of the study, as well as a decrease from No Interaction 1→ No Interaction 2 within phases. This may reflect higher anticipation and alertness at the beginning of training sessions and a lesser need for attention and cortical resources at the end of training.
- Subject 1 displayed an increase in theta and decrease in alpha power across No Interaction 1 which supports the claim that there was more alertness at the beginning of training sessions [39,40].
- Subject 2 displayed a suppression in theta and increase in alpha power from baseline to intervention phases across No Interaction 1 and No Interaction 2, as well as within the B2 phase. This theta-alpha oscillation has been associated with an increase in cognitive performance and suggests that power mobility training was somewhat responsible for short-term and long-term increases in cognitive performance [40].
- All 3 subjects exhibited substantial rehabilitative growth that appeared to correlate well with trends seen in the EEG metrics.

2.4.4 Limitations and Future Work

Several of the limitations observed during the exploratory study still carry over to this study. Because the data acquisition system was customized from the original wireless EMOTIV® EPOC neuroheadset, the changes could have introduced some bias. The selection of gold cup disc electrodes and skin prep gel rather than Ag/AgCl electrodes or moistened felt pads for example, could have affected signal strength and quality. Additionally, a short preparation time was needed in order to be tolerated by the subject, so the subject's hair was moistened before administering the cap to create a better connection from the electrodes to the scalp [1]. This introduced cross bridges which are known to cause low frequency artifacts below 1 Hz. Although a high pass filter administered a 2 Hz cut-off, the cross bridges may have still caused some imbalance in signal strength among channels [1].

Another issue with the acquisition system, and with EEG in general was artefactual contamination caused by the subject's head motion and wheelchair vibration. Artifacts were removed both manually and using automated wICA analysis, but more processing such as visual inspection of independent components may be needed to further remove ocular artifacts..

Data quality was also limited by low temporal and spatial resolution. EEG data were sampled at 2048 Hz but down sampled to 128 Hz for wireless transmission, whereas most modern systems have a sampling rate of 500 Hz or more [49]. Additionally, only 14 recording electrodes were used, which is lower than more typical systems which generally use a minimum of 32 channels. Because of this low spatial resolution, high specificity cannot be attained, and changes that occur within individual electrodes are likely to reflect the electrophysiology of a more global region. While high specificity was not required for this study, this may need to be rectified in future work if individual neuronal networks are to be realized.

The single-subject A-B-A-B design also led to some statistical issues. Because analysis is limited to each subject, there was a maximum of 5 trials per phase. Although this was only a pilot project, this small sample size could have a large effect on the outcome of the analysis since EEG is very sensitive to variations in the experimental setup and subject's emotional state throughout the study. An obvious solution to this would be to either increase the number or length of the phases but this is more difficult in practice with this study population. Increased training intervals or length could cause a greater strain on both the subjects and their parents. A longer study length could also start to reflect changes in the EEG due to the plasticity of the developing brain more so than power mobility training [1]. In its current state, more subjects would be needed to validate findings in the SSRD-A-B-A-B design.

Moving forward, a more robust data acquisition system with a higher sampling frequency and more electrode channels would be invaluable. While we were able to observe certain trends in the data using the current system, better spatial and temporal resolution would help to validate our results. The study design might need to be revised and more subjects recruited in order to account for the low number of samples per trial and strengthen statistical significance. The use of the four metrics proved helpful in providing more objective measures to examine changes in the brain, but more extensive analysis of PSD and MSC should be performed. The spectral range should be extended through the gamma frequency as this may provide some measure of higher order thought. Moreover, PSD and MSC should be further divided into upper and lower alpha, as literature suggests upper alpha may provide insight into cognitive functioning [39]. MSC should also be extended to the beta and gamma ranges in tandem with the PSD. Investigation of mu rhythm, a lesser known EEG oscillation originating from the central region of the scalp, may provide additional insight into cortical activity of the brain. The few studies that have explored mu activity have shown that it lies in the same frequency range as the upper alpha band and is concentrated entirely over the sensorimotor cortex [64-66]. These studies suggest that mu reflects the activity of neurons that play a critical role in imitation learning and understanding the actions of others [64]. Analysis of this rhythm could potentially influence how power mobility is initially taught, along with subsequent training tasks. Because mu oscillations are believed to last for a very brief period (0.5-2 seconds), incorporation of event-related synchronization and desynchronization measurements would need to occur, which may be difficult with this study population [64-66].

MI analysis in the presence of TE was deemed superfluous for this study, as TE incorporated measures of MI to determine the directionality of information transfer. Because computation of TE is fairly time consuming however, MI could be useful when the number of subjects is larger, thus cutting down on computational strain. MI would be used to reveal regions of shared information so that subsequent TE computations could be limited to those specific areas, rather than across all electrode pairs. Incorporation of these changes would provide a more statistically sound and comprehensive view into how power mobility training affects the brain in children with severe disabilities.

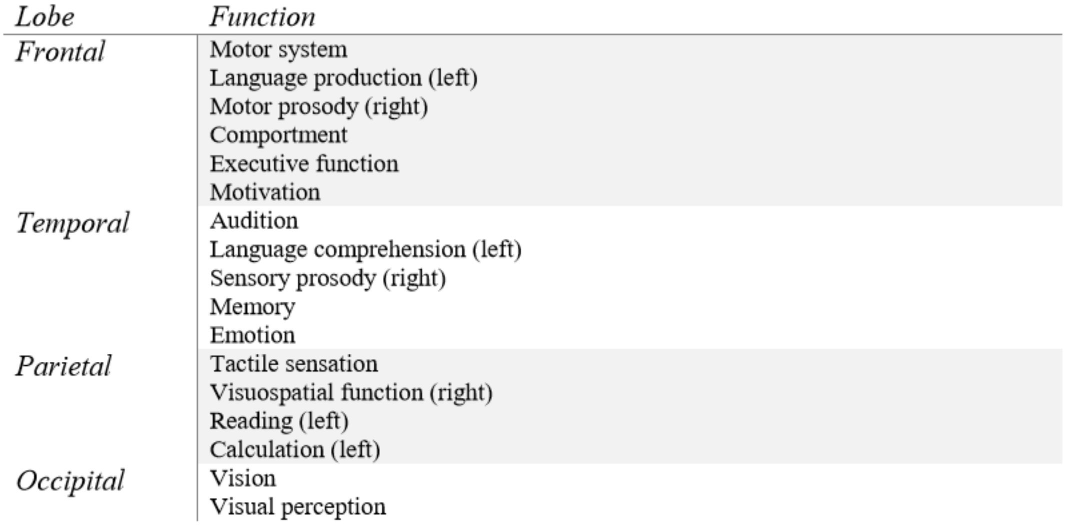
3 Extended Review of Literature and Extended Methodology

3.1 Extended Review of Literature

3.1.1 Anatomy of the Brain

The brain consists of three main regions: the hindbrain, the midbrain, and the forebrain. The hindbrain includes the cerebellum, which coordinates movement, and the brainstem, which is responsible for many of the body's vital functions such as respiration and heart rate. The uppermost region of the brainstem is the midbrain, which controls certain reflexes and is part of the circuitry for eye and other voluntary movements [9,10]. Lastly, the forebrain consists mainly of the cerebrum and "inner brain", a region of structures that lie underneath the cerebrum. The cerebrum is the largest part of the brain and has a surface that contains many ridges called gyri that are characterized by fissures, or sulci [9]. The pattern of gyri and sulci vary widely between individuals as well as between hemispheres of the same brain. The surface of the cerebrum is known as the cerebral cortex and is organized into units of interconnected columns comprised of six sublayers of cells. The cerebrum is split into two hemispheres which are further divided into specialized lobes that are responsible for highly complex functions. The lateral sulcus divides the frontal and temporal lobes while the central sulcus divides the frontal from the parietal lobe [9]. There is no sulcus that defines the boundary for the occipital lobe [9]. Table 4 shows the 4 lobes and their respective functions.

Table 4. Brain regions and their functions [1].



Lobe	*Function*
Frontal	Motor system
	Language production (left)
	Motor prosody (right)
	Comportment
	Executive function
	Motivation
Temporal	Audition
	Language comprehension (left)
	Sensory prosody (right)
	Memory
	Emotion
Parietal	Tactile sensation
	Visuospatial function (right)
	Reading (left)
	Calculation (left)
Occipital	Vision
	Visual perception

The parietal lobe is associated with integrating sensory information while the occipital lobe processes visual information. The temporal lobe receives auditory signals and discerns languages, while the frontal lobe is responsible for higher order thought and processing, as well as speech and motor activity.

3.1.2 Signaling in the Brain

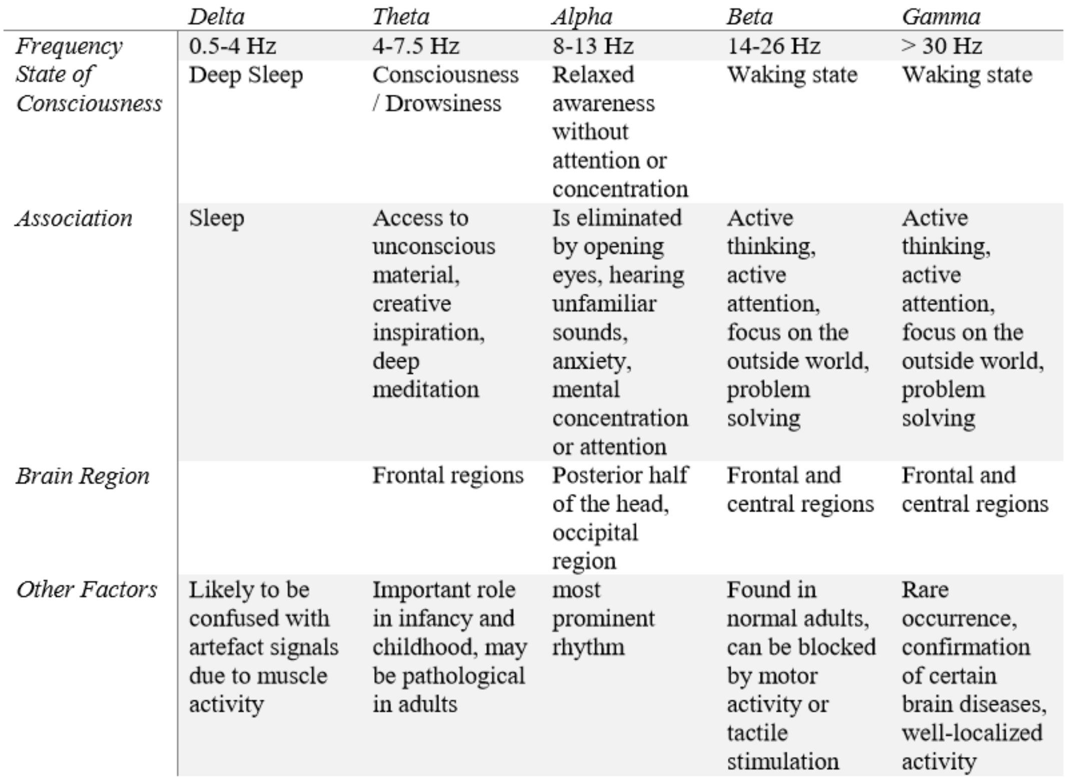
The brain is composed of functional units called neurons. Almost all processes are controlled by the signaling that occurs between neurons located in all parts of the body. A neuron can be divided into three distinct parts: the soma, dendrites, and axon. The soma is the cell body that contains the nucleus and ribosomes that are essential in manufacturing the molecules that are necessary to keep the cell alive and functioning. The dendrites extend from the soma and are responsible for receiving incoming sensory information and transferring it to the integrating regions of sensory neurons. The axon consists of the nerve fiber and the axon terminal and is the site of signal propagation towards target cells. The cortex houses approximately 70% of the 100 billion neurons which form the grey matter (neuronal somas) seen on the outer rim of the cerebral hemispheres along with the white matter (neuronal axons) on the inner area [9].

Signaling occurs when a stimulus causes a change in the permeability and concentrations of sodium (Na^+), potassium (K^+), and chlorine (Cl^-) in a cell, which alters the resting membrane potential (~ -70 mV) [10]. The intracellular space is comprised mainly of K^+ and anions while the extracellular space is filled with Na^+ and Cl^-. The stimulus causes the local sodium channels in the membrane to open, allowing an influx of sodium into the cell. This causes the membrane potential in that region to become more positive (depolarize). If the membrane potential reaches a threshold (~ -40 mV), Na^+ rapidly floods the cell due to an increase in sodium conduction. This flow of positive charge into the cell causes a rapid depolarization. Following this depolarization, Na^+ channels close and K^+ channels open, allowing potassium to flow out of the cell while preventing sodium from entering. The exodus of positive charge causes the membrane potential to decrease (repolarization). Because the potassium channels are slow to close and return to normal levels of conductance, extra K^+ flows out of the cell, causing the membrane potential to become more negative than its original resting potential (hyperpolarization). Eventually, the Na^+/K^+ pump reestablishes the local membrane resting potential by restoring the extra/intracellular concentrations of the ions. This cycle of events is known as an action potential (AP). This AP is able to travel down the length of an axon because the local ion currents prompt more voltage-gated channels to open further down the fiber, repeating the cycle. The action potential will eventually reach the axon terminal that when stimulated by the currents, will either release or inhibit neurotransmitters. These neurotransmitters, such as dopamine and epinephrine, are chemical substances that diffuse across synapses or junctions and transfer the action potential to a recipient cell (nerve, muscle, organ, etc.) so that it may carry out its own impulse-induced actions [10].

3.1.3 Electroencephalography (EEG)

The neurons in the cortex are somewhat synchronized in regular firing rhythms which allows for meaningful recording of the electrical activity, known as an electroencephalogram (EEG). The EEG is a summation of the excitatory and inhibitory post-synaptic potentials (EPSP & IPSP) of cortical neurons whose large-scale activity can reflect information on the various states and characteristics of the brain. The EEG waveform consists of different frequency waves that can be extracted to provide insight on different brain activities. These rhythms can be seen in Table 5.

Table 5. Brain rhythms and their associated states of consciousness [1].



	Delta	*Theta*	*Alpha*	*Beta*	*Gamma*
Frequency	0.5-4 Hz	4-7.5 Hz	8-13 Hz	14-26 Hz	> 30 Hz
State of Consciousness	Deep Sleep	Consciousness / Drowsiness	Relaxed awareness without attention or concentration	Waking state	Waking state
Association	Sleep	Access to unconscious material, creative inspiration, deep meditation	Is eliminated by opening eyes, hearing unfamiliar sounds, anxiety, mental concentration or attention	Active thinking, active attention, focus on the outside world, problem solving	Active thinking, active attention, focus on the outside world, problem solving
Brain Region		Frontal regions	Posterior half of the head, occipital region	Frontal and central regions	Frontal and central regions
Other Factors	Likely to be confused with artefact signals due to muscle activity	Important role in infancy and childhood, may be pathological in adults	most prominent rhythm	Found in normal adults, can be blocked by motor activity or tactile stimulation	Rare occurrence, confirmation of certain brain diseases, well-localized activity

The delta wave (0.5-4 Hz) is the dominant rhythm in sleep stages three and four and tends to have the largest amplitude of all the brain waves. It is also thought to be associated with continuous-attention. The theta wave (4-8 Hz) has a rhythm that is normal during sleep and can be seen in awake children but not adults. It's often split into two categories; 4-5.45 Hz is associated with decreased arousal and increased drowsiness, while 6-7.45 Hz is enhanced during tasks involving working memory. The alpha wave (8-13 Hz) is most prominent when the eyes are closed and the person is relaxed. It is suppressed when the eyes are open and by mental effort such as calculations or concentration. It is thus considered to indicate the level of cortical activation (more activation, less alpha). The beta wave (13-30 Hz) is prominent in alert individuals whose eyes are open. Lastly, the gamma wave's significance (30-50 Hz) is not well understood, but is thought to be associated with higher mental activity such as perception and consciousness.

3.1.4 Network Connectivity

A network is a mathematical representation of a complex system defined by a collection of nodes and the links between pairs of nodes [22,23]. Nodes as they pertain to the brain, are generally brain regions while the links between them could be anatomical, functional, or effective. Anatomical connections are usually characterized by white matter as the axonal tracts are responsible for the transmittance of signals from neuron to neuron. Functional connectivity however, corresponds to the magnitudes of temporal correlations in activity and can occur between anatomically unconnected regions [22]. Effective connections are a representation of causal influences of one region on another which are estimated in a variety of ways [22]. Figure 3-1 below gives a pictorial representation of connectivity.

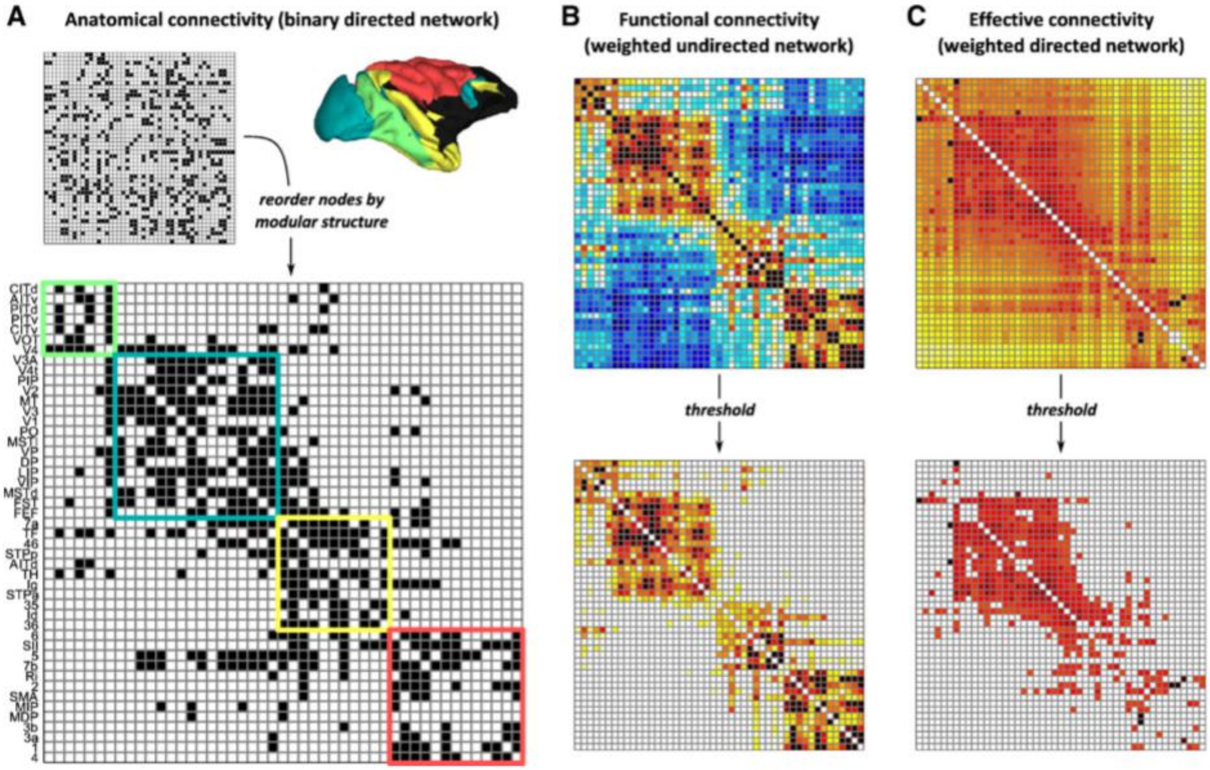


Figure 3-1 (a, b, c). Connectivity is shown in 3 different categories. Anatomical connectivity (A) is shown where network modules representing groups of somatomotor regions are color-coded by anatomical regions. Functional Connectivity (B) represents the cross-correlation of the regional signal, estimated from simulated model dynamics. Warm colors represent positive correlations and cooler colors represent negative correlations. The bottom figure in B shows the same connectivity but with negative correlations zeroed out. The effective connectivity (C) was created by computing inter-regional transfer entropy. The bottom figure in C has a threshold applied to show the strongest connections [22].

Early researchers of cerebral cortex anatomy formed the foundation of neural network connectivity, referring to the concept of nodes as localization within the cortex. Brodmann, for which the term “Brodmann areas” are named after, employed histological dissection and cytoarchitectonics to first characterize anatomical localization within the brain [21].

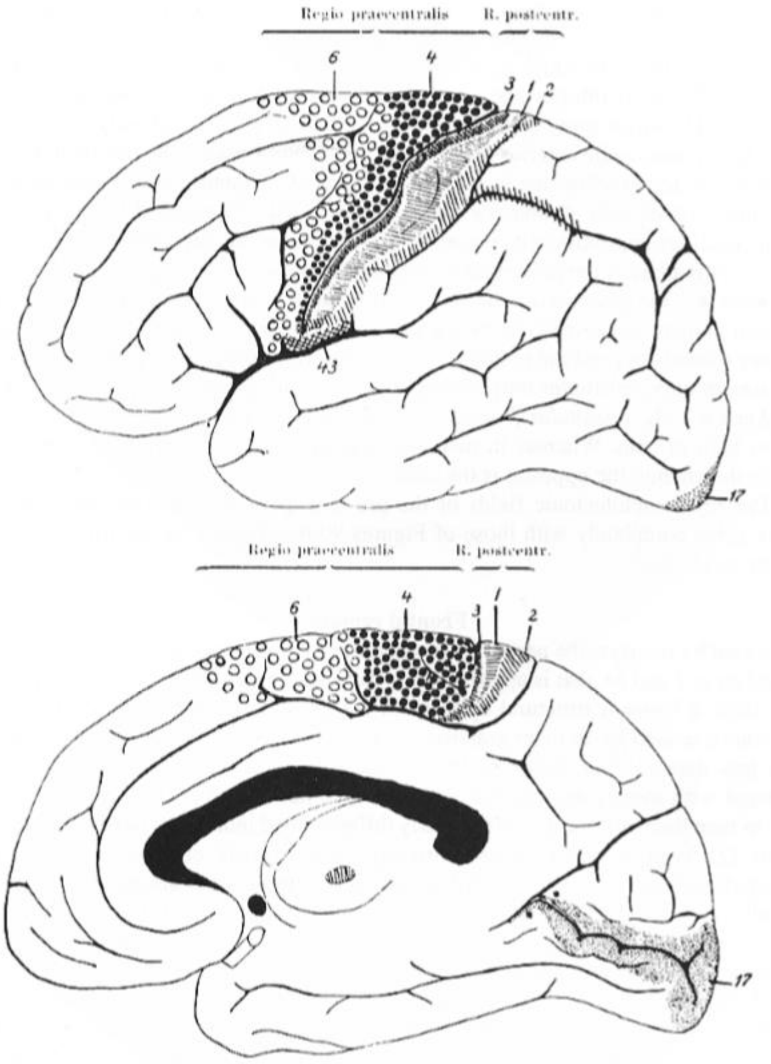


Figure 3-2. Early depictions of Brodmann areas characterized only by anatomical localization in the brain [21].

As seen in Figure 3-2, areas of the brain are numbered according only to neuronal organization. Since his publications during the early 20th century, many others have expanded upon his findings, refining the precision of the original anatomical locations and correlating them to a diverse range of cortical function.

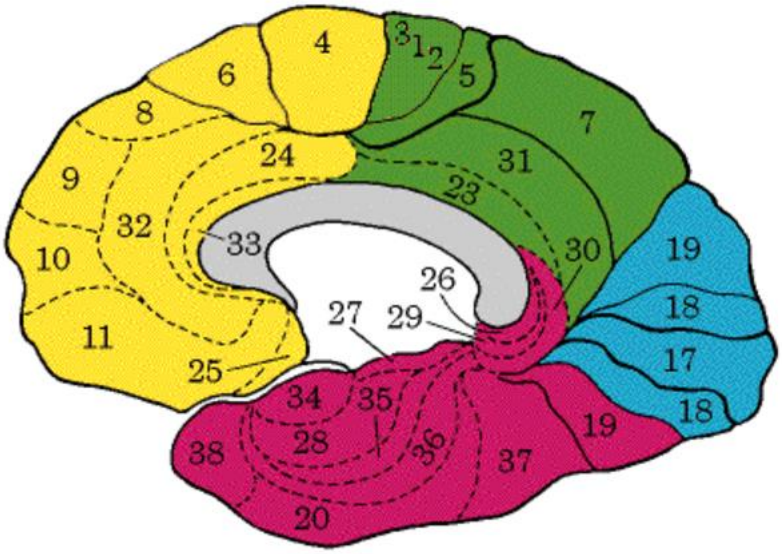
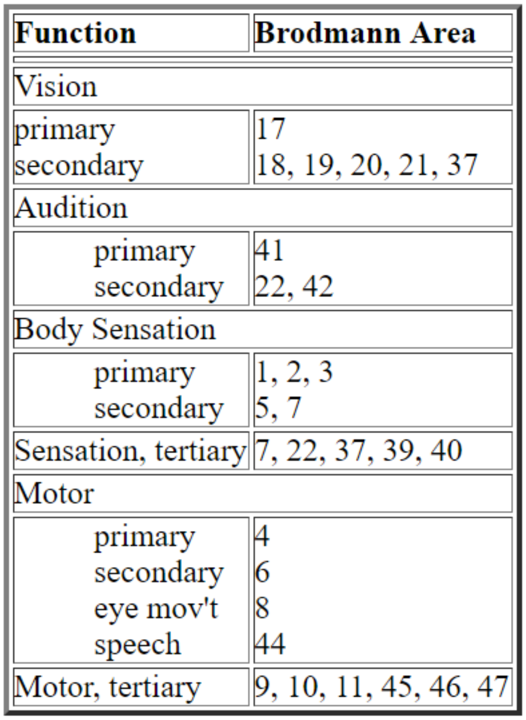


Figure 3-3. Modern depictions of Brodmann areas characterized by both anatomy and function [24].

Table 6. Modern Brodmann areas and their functions [24].



Function	**Brodmann Area**
Vision	
primary	17
secondary	18, 19, 20, 21, 37
Audition	
primary	41
secondary	22, 42
Body Sensation	
primary	1, 2, 3
secondary	5, 7
Sensation, tertiary	7, 22, 37, 39, 40
Motor	
primary	4
secondary	6
eye mov't	8
speech	44
Motor, tertiary	9, 10, 11, 45, 46, 47

Collaborators at the Brainnetome Project use multimodal magnetic resonance imaging (MRI) to ascertain input and output patterns of each cortical area to provide a comprehensive atlas of connectivity [25]. Undertakings such as these allow for a richer understanding of how signals are carried within the brain and how systems such as sensory integration and motor capacity are all interrelated. In general, information tends to flow from the back of the brain to the front as information is received and processed, and decisions are made [24].

3.1.5 Information Theory

Information theory is concerned with the analysis of a communication system. It attempts to construct a mathematical model to indicate the level in which a “receiver” is able to accurately receive and replicate transmitted information from a “source” via a channel [13]. The source refers to the person, machine, or in this case, region of the brain, that produces the information to be communicated [13]. An encoder associates each of these messages with an “object” that is suitable for transmission over a given channel, i.e. binary digits or a continuous wave. A channel is defined as the medium over which a message is transmitted. In an effort to understand information theory, I employ an analogy similar to the one suggested in Mcdonnell et al [14]. Imagine one person is trying to send a series of messages to a second person on the opposite side of a brick wall. Assume the second person is not allowed to speak or communicate in any way to the first person [14]. How should the first person send each message?

There are three parameters that the first person can modulate in order to increase the chance that the second person will understand their message. The first parameter is volume, i.e. shouting louder. Perhaps there is background noise around the brick wall so shouting louder would allow the more important message to be heard above the noise. This is referred to as transmission power. The second parameter is speed, i.e. how fast the person is talking. Person B has a greater chance of hearing every word if Person A speaks slower. This is known as transmission rate. The third parameter is redundancy, i.e. the number of ways a person can relay a central message. One approach is to use more words to describe the concept or elaborate on a message to increase the chance that the concept is understood. For example, let's say the message is "the large ball is red". Person A could attempt to relay this message by elaborating on each of the different characteristics of the ball, e.g. "the giant, humongous, big ball is red", or "the large spherical, round object is red", or "the large ball is the same color as a stop sign and the first color in a rainbow". All descriptions of the message still relay the central statement that the large ball is red. Another method is to simply repeat the same message over and over. These two approaches are known as coding. Claude Shannon, who originally proposed the idea of information theory, realized that the way in which one codes the message allows for the information to be transmitted with a marginal probability of error given a finite transmission power and rate even in the presence of noise [15]. His reasoning was that each message has a meaning, and that if one codes each meaning differently, it makes it much simpler for a system to select the correct meaning from a set of possible messages with increasing accuracy [15].

Information theory characterizes random processes (e.g. EEG data) using probability density functions to estimate the amount and direction of information being transmitted within the given system. In the context of information theory, information is thought of abstractly as the resolution of uncertainty. Therefore, when referring to the information of a random variable, the metric being alluded to is Shannon entropy. This entropy describes the reduction of uncertainty in predicting future values of that variable that occurs when one measures the current value of the variable [29]. This approach can be extended to two variables by observing the casual dependency between them. Wiener defined this dependency based on an increase of prediction power, i.e. a signal X is said to cause a signal Y when the future of signal Y is better predicted by adding knowledge from the past and present of signal X than by using the present and past of Y alone [59]. Thus, when prediction enhancement can be associated with uncertainty reduction, measures of causality can be derived from information theoretic concepts. Frequently used information theoretic measures include transfer entropy and mutual information [15-18]. Derivations of these measures can be found in the following section. Information theory has a wide array of applications, particularly in neuroscience. Rapaport and Horvath [19] showed that neurons are essentially information channels that are able to transmit large quantities of information on the level of ~4000 bits/second during continuous time interval coding. Recent advancements in information theory have allowed researchers to estimate directed information in order to investigate causal relationships in neuron networks and neural spike trains [20]. Additionally, others have used information theory measures to improve connectivity in both linear and nonlinear interactions in magnetoencephalography (MEG) recordings during motor tasks, as well as in modeling retinal firing patterns [16]. Information theory is becoming a more widely utilized tool in both biological and neuroscience research as measuring and interpreting its applicable quantities becomes better understood.

3.2 Extended Methodology

3.2.1 Data Acquisition System

The acquisition system was very similar to that used in the exploratory study [1]. The EEG data was collected using a modified EMOTIV® EPOC neuroheadset with 14 recording electrodes. It was hypothesized that the original placement of the electrodes did not adequately capture the majority of the EEG signal because of their location on the outer rim of the scalp [1]. Therefore, the configuration was modified to incorporate more frontal and parietal electrodes in the center of the scalp.

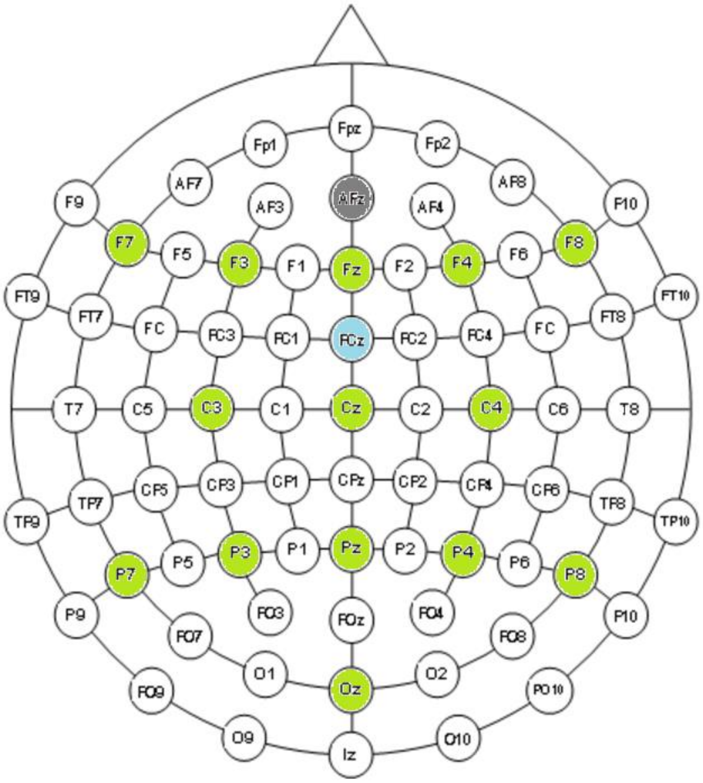


Figure 3-4. New electrode placement according to the International 10-20 system. The 16 electrodes include front (F), central (C), and parietal (P) channels, two reference electrodes (AFz & FCz). Even numbers refer to electrodes on the right hemisphere and odd numbers refer to those on the left hemisphere.

The electrodes in Figure 3-4 that are highlighted green are recording electrodes (F7, F5, Fz, F4, F8, C3, Cz, C4, P7, P5, Pz, P4, P8, Oz), while the two electrodes highlighted in gray and blue are reference electrodes (AFz, FCz). The same wireless acquisition system was also used for this study [1].

3.2.4 Coherence Analysis

Coherence is a measure of correlation between the frequency components of two signals and can be used to determine the constancy of a linear phase relationship between the Fourier components of the corresponding signals [11]. It is often viewed as the frequency domain analog of the squared correlation coefficient (r^2) [11,12]. Coherence is defined as

$$C_{xy}(t,f) = \frac{S_{xy}(t,f)}{\left[S_x(t,f)S_y(t,f)\right]^{1/2}} \quad , \tag{5}$$

where S_{xy} is the cross-spectral density and S_x and S_y are the auto-spectral densities of *X* and *Y* respectively. To calculate the auto-spectral densities, Welch’s method was used and can be defined as

$$\hat{P}_x^{(k)}(\omega) = \frac{1}{N}\left|\sum_{n=0}^{N-1} v[n]x^{(k)}[n]e^{-j\omega n}\right|^2, \qquad k = 1, \ldots, K \quad , \tag{5.1}$$

where $\omega = 2\pi f_s$, K was the number of overlapping segments (usually 50% overlap), and v[n] was a windowing function (e.g. Hamming). Stationarity was assumed within each segment. The estimate of the PSD was then given by

$$\hat{S}_x(\omega) = \frac{1}{K}\sum_{k=1}^{K} P_x^{(k)}(\omega) \quad . \tag{5.2}$$

Through ensemble averaging over multiple periodograms, the bias and variance in the shorter epochs was greatly reduced to provide a smoother, more stable power spectrum estimation. The respective spectrograms, S_x and S_y, for the data were found by calculating the square of the sum of the segments, as seen in Eq. 5.2. S_{xy} was computed in a similar fashion, ensemble averaging the windowed cross correlation of *X* and *Y*. The auto and cross correlation periodograms were then used to compute the magnitude-squared coherence (MSC), $C_{xy}(t,f)^2$, derived from Eq. 5. When components share a constant phase relationship, the MSC will approach one, whereas the MSC for uncorrelated phases (regions containing white noise), will approach zero.

3.2.2 Mutual Information Analysis

Mutual information (MI) is a quantity that measures how much information is communicated in one random variable about another. Maximum MI is achieved when two time series are the same, while minimum MI is achieved when the two series are statistically independent. MI allows for the estimation of both the linear and nonlinear statistical dependencies between time series [26]. For a measurement x_i generated from system *X* and measurement y_i generated from system *Y*, the mutual information is the amount of information that measurement x_i provides about y_i and can therefore be viewed as the dynamical coupling or information transmission between *X* and *Y* [27]. The expected (average) amount of information obtained from a given observation of X is defined as the Shannon entropy H of a system, denoted as

$$H(X) = -\sum_{x_i} P_X(x_i)\log_2 P_X(x_i) \quad , \tag{6}$$

where $P_x(x)$ is the normalized histogram of values observed for the measurement *x* and $P_x(x_i)$ is the probability that an isolated measurement will find the system in the *i*th element of the histogram bin [27]. Before any measurements of *X* are taken, *H* is known as uncertainty. The addition of system *Y* introduces conditional uncertainty in *X* so when $Y = y_i$, *H(x)* is now replaced with

$$H(X|Y=y_j) = -\sum_{x_i} \frac{P_{XY}(x_i, y_j)}{P_Y(y_j)} \log_2 \frac{P_{XY}(x_i, y_j)}{P_Y(y_j)}$$,

(6.1)

where $P_{XY}(x_i,y_i)$ is the joint probability of measurements of *X* and *Y* that produce the values *x* and *y*. Equation 6.1 indicates the amount of uncertainty in a measurement of *x*, given that a measurement of *y* was found to be y_j. From this, the mean conditional uncertainty on *X* over y_j under the condition that *Y* is known, is given by

$$H(X|Y) = \sum_{y_j} P_Y(y_j) H(X|Y=y_j)$$

(6.2)

$$= -\sum_{x_i,y_j} P_{XY}(x_i, y_j) \log_2 [P_{XY}(x_i, y_j)/P_y(y_j)]$$

(6.3)

$$= H(X,Y) - H(Y)$$,

(6.4)

where

$$H(X,Y) = -\sum_{x_i,y_j} P_{XY}(x_i, y_j) \log_2 [P_{XY}(x_i, y_j)]$$

(6.5)

The conditional entropy measures the amount of uncertainty in *X* given that a signal *Y* is known. Because the a priori uncertainty on *X* is *H(X)*, and the a posteriori uncertainty on *X* given a measurement of *y* is *H(X|Y)*, the amount that a measurement of *y* reduces the uncertainty of *x* can be given by

$$I_{XY} = H(X) - H(X|Y)$$

(6.6)

$$= H(X) + H(Y) - H(X, Y)$$

,

(6.7)

which can be rewritten as

$$I_{XY} = \sum_{x_i, y_j} P_{XY}(x_i, y_j) \log_2 \frac{P_{XY}(x_i, y_j)}{P_X(x_i) P_Y(y_j)}$$

.

(6.8)

This is the mutual information between measurements of *X* and *Y* which predicts the average number of bits (based on a $\log_2$ derivation) that can be predicted about *x* given a measurement of *y* [27,28]. If *X* is independent of *Y*, $P_{XY}(x_i,y_j)$ factorizes to $P_X(x)P_Y(y)$ and the mutual information becomes zero. Conversely, higher mutual information indicates that two signals contain more information about each other, signifying that they are more closely related. MI is therefore used in this study to investigate the amount of shared information between pairs of EEG electrodes.

3.2.3 Transfer Entropy Analysis

Transfer entropy (TE) is a non-linear statistic used to measure the amount of directed information transfer between two random processes [29,30]. It is an extension of the Wiener-Granger causality framework and can be based off mutual information measures. TE follows the standard steps of prediction-based causality which are as follows: Imagine we want to determine whether variable *X* causes variable *Y*. The initial step would be to try and predict the current value of *Y* using previous values of *Y*. The next step would be to perform another prediction of the current value of *Y* using historical values of both *X* and *Y*. Lastly, you would compare the two predictions. If the latter prediction is deemed better or more reliable than the former, one can conclude that *X* is somewhat responsible for causing *Y*. There are several methods to compute

TE, including using MI-based measures. Given the conditional entropy defined in Eq. 6.6, transfer entropy from *X* to *Y* can be defined as follows:

$$T_{X\to Y} = H(y_i|y^{(l)}_{i-t}) - H(y_i|y^{(l)}_{i-t},\ x^{(k)}_{i-\tau})$$

(7)

$$= \sum_{y_i, y^{(l)}_{i-t}} p(y_i,\ y^{(l)}_{i-t}) \log \frac{p(y^{(l)}_{i-t})}{p(y_i, y^{(l)}_{i-t})} - \sum_{y_i, y^{(l)}_{i-t}, x^{(k)}_{i-\tau}} p(y_i,\ y^{(l)}_{i-t},\ x^{(k)}_{i-\tau}) \log \frac{p(x^{(k)}_{i-\tau}, y^{(l)}_{i-t})}{p(y_i, y^{(l)}_{i-t}, x^{(k)}_{i-\tau})} ,$$

(7.1)

where *i* indicates a given point in time, *τ* and *t* are time lags for *X* and *Y* respectively, and *k* and *l* are the block lengths of past values of *X* and *Y* respectively. The past values for *x* and *y* are given as $x(k)_{i-\tau} = \{x_{i-\tau-k+1}, x_{i-\tau-k+2}, \dots , x_{i-\tau}\}$, and $y(l)_{i-t} = \{y_{i-t-l+1}, y_{i-t-l+2}, \dots , y_{i-t}\}$. Since

$$p(y_i,\ y^{(l)}_{i-t}) = \textstyle\sum_{x^{(k)}_{i-\tau}} p(y_i,\ y^{(l)}_{i-t},\ x^{(k)}_{i-\tau}) ,$$

(7.2)

the two arguments in the separate summations can be combined, yielding

$$T_{X\to Y} = \sum_{y_i, y^{(l)}_{i-t}, x^{(k)}_{i-\tau}} p(y_i,\ y^{(l)}_{i-t},\ x^{(k)}_{i-\tau}) \log \frac{p(y^{(l)}_{i-t}) p(y_i, y^{(l)}_{i-t}, x^{(k)}_{i-\tau})}{p(y_i, y^{(l)}_{i-t}) p(y^{(l)}_{i-t}, x^{(k)}_{i-\tau})}$$

(7.3)

$$= \sum_{y_i, y^{(l)}_{i-t}, x^{(k)}_{i-\tau}} p\left(y_i,\ y^{(l)}_{i-t},\ x^{(k)}_{i-\tau}\right) \log \frac{p\left(y_i|y^{(l)}_{i-t}, x^{(k)}_{i-\tau}\right)}{p\left(y_i|y^{(l)}_{i-t}\right)} .$$

(7.4)

4 Appendix

4.1 Code: Preprocessing for Artifact Removal (GUI)

```
function varargout = EEG_GUI_ArtRemTest(varargin)
% EEG_GUI_ARTREM MATLAB code for EEG_GUI_ArtRem.fig
%      EEG_GUI_ARTREM, by itself, creates a new EEG_GUI_ARTREM or raises the existing
%      singleton*.
%
%      H = EEG_GUI_ARTREM returns the handle to a new EEG_GUI_ARTREM or the handle to
%      the existing singleton*.
%
%      EEG_GUI_ARTREM('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in EEG_GUI_ARTREM.M with the given input arguments.
%
%      EEG_GUI_ARTREM('Property','Value',...) creates a new EEG_GUI_ARTREM or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before EEG_GUI_ArtRem_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to EEG_GUI_ArtRem_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help EEG_GUI_ArtRem

% Last Modified by GUIDE v2.5 21-Jan-2016 14:06:29

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @EEG_GUI_ArtRem_OpeningFcn, ...
                   'gui_OutputFcn',  @EEG_GUI_ArtRem_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before EEG_GUI_ArtRem is made visible.
```

```
function EEG_GUI_ArtRem_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to EEG_GUI_ArtRem (see VARARGIN)

% gabby's head radius is 6.266 cm

clc;
setappdata(0,'hMainGUI',gcf);
setappdata(gcf,'mainHandles',handles);
initializeGUI(handles);

% Choose default command line output for EEG_GUI_ArtRem
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes EEG_GUI_ArtRem wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = EEG_GUI_ArtRem_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes when figure1 is resized.
function figure1_ResizeFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% ========================================================================
% SLIDERS
% ========================================================================

% --- Executes on slider movement.
function slider_display_Callback(hObject, eventdata, handles)
% hObject    handle to slider_display (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider

updateGUI(handles);
```

```
% --- Executes during object creation, after setting all properties.
function slider_display_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider_display (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function slider_epoch_Callback(hObject, eventdata, handles)
% hObject    handle to slider_epoch (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider

global Fs

% get epoch length from slider
eLength = floor(str2double(get(handles.slider_epoch,'String')));

% set edit value for epoch to epoch length
set(handles.edit_length,'String',sprintf('%.1f',eLength/Fs));

% refresh GUI
updateGUI(handles);

% --- Executes during object creation, after setting all properties.
function slider_epoch_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider_epoch (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% ========================================================================
% PUSHBUTTONS
% ========================================================================

% --- Executes on button press in pushbutton_load.
function pushbutton_load_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_load (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global rawdata cleandata file icaEEG wicaEEG artdata filename resetdata ...
```

```
    icaComp Fs

% indicate loading status
set(handles.text_loading,'String','Loading...');
dataLoaded = 0; % reset flag for loaded data

% load file selected by user
[filename,filepath,filteridx] = uigetfile({'*.edf','EDF-files (*.edf)';...
    '*.csv','CSV-files (*.csv)';...
    '*.mat','MAT-files (*.mat)';},'Select Data File');
file = strcat(filepath,filename); % concatenate file path and name

% load data depending on file format
switch filteridx
    case 0
    case 1 % loading edf file
        % reset raw data
        rawdata = [];

        % read edf file
        rawdata = readEDF(file);    % read edf file
        rawdata = [rawdata{:}];
        rawdata = rawdata(Fs+1:end-Fs,3:16)';
        %rawdata = rawdata(1, [3:16 Fs+1:end-Fs]);
        %rawdata = rawdata([3:16 36],Fs+1:end-Fs); % extract channels of interest
        % 3-16 eeg channels, 36 marker, remove first and last second from data

        if size(rawdata,1)<15
            n = 15-size(rawdata,1); % calculate missing channels
            rawdata = [rawdata; zeros(n,size(rawdata,2))];
            % add zeros for missing channles
        end
        % set flag to data loaded
        dataLoaded = 1;

        % display file name on GUI
        set(handles.text_file,'String',filename);

        % indicate end of loading process
        set(handles.text_loading,'String','Done.');
    case 2 % load csv file
        % reset rawdata
        rawdata = [];

        % read csv file
        rawdata = xlsread(file);
        rawdata = rawdata(:,[3:16 36]); % extract channels of interest
        % 3-16 eeg channels, 36 marker, remove first and last second from data

        if size(rawdata,1)<15
            n = 15-size(rawdata,1); % calculate missing channels
            rawdata = [rawdata; zeros(n,size(rawdata,2))];
            % add zeros for missing channles
        end
        % set flag to data loaded
        dataLoaded = 1;

        % display file name on GUI
        set(handles.text_file,'String',filename);

        % indicate end of loading process
        set(handles.text_loading,'String','Done.');
    case 3
        % reset raw data
        rawdata = [];

        % load variable
        vars = load(file);
        varname = fieldnames(vars); % get variable name
        load(file);

        % read data from variable
        rawdata = eval(char(varname{1}));

        % add marker channel if channel # smaller than 15
        %rawdata = rawdata(1:14,:);
        if size(rawdata,1)<15
            n = 15-size(rawdata,1); % calculate missing channels
            rawdata = [rawdata; zeros(n,size(rawdata,2))];
            % add zeros for missing channles
        end

        % set flag to data loaded
        dataLoaded = 1;

        % display file name on GUI
        set(handles.text_file,'String',varname);

        % indicate end of loading process
        set(handles.text_loading,'String','Done.');

end


if (dataLoaded)
    % search for first marker in data
    ind = find(rawdata(15,:)>0,1,'first');
    %ind = find(rawdata > 0);
    if isempty(ind) % if no marker set indices to one
        ind = 1;
    end
    % trim data to first marker
    rawdata = rawdata(:,ind:end);

    % get only eeg data
    data = rawdata(1:14,:)';
    % remove DC value
    data = detrend(data,'constant');
    % highpass at 2Hz
    [b,a] = butter(7,[2/64],'high');
    filtdata = filtfilt(b,a,data);

    % average reference
    set(handles.text_process,'String','status: AvgRef processing...');
    avgRefEEG = avgReference(filtdata');

    % add markers
    rawdata = [avgRefEEG; rawdata(15,:)];
    % save data in variables
    cleandata = rawdata;
    resetdata = cleandata; % save this data as reset data
    % reset variables
    artdata = nan(size(rawdata));
    icaEEG = zeros(size(rawdata));
    wicaEEG = zeros(size(rawdata));

    % indicate statuse on GUI
    set(handles.text_process,'String','status: AvgRef done.');

    % switch to clean data view
    set(handles.popupmenu_view,'Value',2);

    % refresh GUI
    updateGUI(handles);
else
    % if no data loaded just refresh GUI with current data
    updateGUI(handles);
end

% --- Executes on button press in pushbutton_video.
function pushbutton_video_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_video (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global vidFile vidLoaded videoIdx

% ask user for directory of video file
[filename,filepath,filteridx] = uigetfile({'*.avi','AVI' },'Select Data File');

% concatenate file path and file name
file = strcat(filepath,filename);

% read video file
vidFile = VideoReader(file);

% get number of frames
nFrames = vidFile.NumberOfFrames;

% set video frame index
videoIdx = 1:nFrames;
```

```
% set flag that video loaded
vidLoaded = 1;

% refresh GUI
updateGUI(handles);

% --- Executes on button press in pushbutton_save.
function pushbutton_save_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_save (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global cleandata filedir comp icaComp Fs wlength

% ask user for directory
filedirnew = uigetdir();

% save path in global variable
if(~isempty(filedirnew))
    filedir = filedirnew;
end

% check if folder for spectra exists, create one if  not
if ~exist([filedir '\spectra'],'dir')
    mkdir(filedir, 'spectra');
end

% check if folder for data exists, create one if  not
if ~exist([filedir '\data'],'dir')
    mkdir(filedir, 'data');
end

% check if folder for components exists, create one if  not
if ~exist([filedir '\components'],'dir')
    mkdir(filedir, 'components');
end

% calculate optimal FFT length
NFFT = 2^nextpow2(wlength+1);

% initalize frequency vector
freq=[];

% get clean data for saving in time domain
savedata = cleandata;

% initalize variable for fft spectrum
savedata_fft=[];

% compute power spectral density with Welch's method, sliding window with
% length of wlength and 50% overlap
for c = 1:size(savedata,1)-1
    [savedata_fft(c,:),freq] = pwelch(savedata(c,:),wlength,wlength/2,NFFT,Fs);
end
```

```
% get onesided spectrum
savedata_fft = savedata_fft(:,1:NFFT/2+1);
% normalize spectrum to total mean power
normval = mean(sum(savedata_fft,2));
savedata_fft = savedata_fft/normval;

% get file name for saving
filenameSave = get(handles.edit_id,'String');

% save timeseries with file name
filedata = [filedir '\data\' filenameSave '_data'];
[pathstr,namedata,ext] = fileparts(filedata);
subsetdata = genvarname(namedata);
eval([subsetdata '= savedata;'])
save(filedata,namedata);

% save PSD with file name
filespec = [filedir '\spectra\' filenameSave '_spec'];
[pathstr,namespec,ext] = fileparts(filespec);
subsetspec = genvarname(namespec);
eval([subsetspec '= savedata_fft;'])
save(filespec,namespec);


% plot spectra for control purposes
figure
plotPSD(savedata_fft,freq)
title('Power Spectrum Density')

% compute spectra of components if ica components available
% resest variable for spectra of ica components
components_fft=[];

if ~isempty(icaComp)
    freq = [];
    k = 1;
    % compute spectra of ica components with Welch's method and window
    % length of wlength and 50% overlap
    for c = comp(end)+1:size(icaComp,1) % iterate of components of interest
        [components_fft(k,:),freq] = pwelch(icaComp(c,:),wlength,wlength/2,NFFT,Fs);
        k = k+1;
    end
    % obtain onesided spectrum
    components_fft = components_fft(:,1:NFFT/2+1);
    freq = freq(1:NFFT/2+1)';
    % normalize to total mean power
    normval = mean(sum(components_fft,2));
    components_fft = components_fft/normval;

    % plot spectrum for control purposes
    figure
    plotPSD(components_fft,freq)
    title('Power Spectrum Density (ICs)')
    % save spectra of ica components
    filecompspec = [filedir '\components\' filenameSave '_comp_spec'];
    [pathstr,namecompspec,ext] = fileparts(filecompspec);
    subsetcompspec = genvarname(namecompspec);
    eval([subsetcompspec '= components_fft;'])
    save(filecompspec,namecompspec);
end


% --- Executes on button press in pushbutton_remove.
function pushbutton_remove_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_remove (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global x Fs rawdata cleandata videoIdx vidFile vidLoaded resetdata

% calculate samples of time selection
xSmpl = round(x*Fs);
xleft = min(xSmpl); % get left border of selection
xright = max(xSmpl); % get right border of selection

% cut and remove selection from raw and clean data
if xright>size(rawdata,2)   % remove data from end
    rawdata = rawdata(:,1:xleft);
    cleandata = cleandata(:,1:xleft);
elseif xleft<1              % remove data from start
    rawdata = rawdata(:,xright:end);
    cleandata = cleandata(:,xright:end);
else
    % remove data in the middle
    rawdata = [rawdata(:,1:xleft) rawdata(:,xright:end)];
    cleandata = [cleandata(:,1:xleft) cleandata(:,xright:end)];

end

resetdata = rawdata;
% cut and remove selection from video if available
if vidLoaded
    xFrames = round(x*vidFile.FrameRate); % calculate frames of selection
    xvidleft = min(xFrames);    % get left border of selection
    xvidright = max(xFrames);   % get right border of selection
    if xvidright>length(videoIdx)   % remove frames at the end
        videoIdx = videoIdx(1:xvidleft);
    elseif xvidleft<1
        videoIdx = videoIdx(xright:end); % remove frames at the beginning
    else
        % remove frames in the middle
        videoIdx = [videoIdx(1:xvidleft+2) videoIdx(xvidright-2:end)];
    end
end

% refresh GUI
updateGUI(handles);
```

```
% --- Executes on button press in pushbutton_reset.
function pushbutton_reset_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_reset (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global resetdata cleandata artdata

% set clean data to reset data
cleandata = resetdata;

% reset detected artifacts
artdata = nan(size(cleandata));

% set display to clean data
set(handles.popupmenu_view,'Value',2);

% refresh GUI
updateGUI(handles);

% --- Executes on button press in pushbutton_test.
function pushbutton_test_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_test (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global rawdata cleandata Fs filename

% intialize variables
rawdata = zeros(15,100*Fs);
cleandata = zeros(15,100*Fs);

% define amplitude
a = -0.5;
b = 0.5;

% generate random noise
rawdata(1:14,:) = a + (b-a).*randn(14,100*Fs);
cleandata(1:14,:) = a + (b-a).*randn(14,100*Fs);

% remove DC component(not necessary actually)
rawdata(1:14,:)  = detrend(rawdata(1:14,:),'constant');
cleandata(1:14,:)  = detrend(cleandata(1:14,:),'constant');

% define a filename
filename = 'test-signal-randn';

% refresh GUI
updateGUI(handles);


% --- Executes on button press in pushbutton_autoDetect.
function pushbutton_autoDetect_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to pushbutton_autoDetect (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global rawdata cleandata artdata Fs concatdata

% get selected datatype
datatype = get(handles.popupmenu_view,'Value');

if datatype==1
    % get rawdata
    data = rawdata(1:14,:);
else
    % get cleandata
    data = cleandata(1:14,:);
end

% initialize variable for detected artifacts
artdata = nan(size(cleandata));

% indicate process status
set(handles.text_process,'String','auto detect: processing...');

% get number of samples
N = size(data,2);

% define length of extension for detection
L = Fs/2;

% calculate standard deviation
sd = std(data,[],2);

% define threshold for detection
thr1 = repmat(4*sd,[1, N]);
% find indices of artifacts based on threshold
[row,col] = find(abs(data) > thr1);
% find indices of clean data
idx = setdiff(1:N,col);
% find indices of artifacts
idx = setdiff(1:N,idx);

% extend regions of artifacts by length L left and right
idx_ext = zeros(1,length(col)*(2*L+1));
for k=1:length(idx)
    idx_ext((2*L+1)*(k-1)+1:(2*L+1)*k) = [idx(k)-L:idx(k)+L];
end
% final indices of signal
idx_sig = setdiff((1:N), idx_ext);
% final indices of artifacts
idx_art = setdiff((1:N), idx_sig);

% remove artifacts - set to zero
artchannel = data;
data(:,idx_art) = 0;
```

```
% mark artifacts for display
artchannel(:,idx_sig) = NaN;
artdata(1:14,:) = artchannel;

% prepare timeseries without artifacts for rejection
concatdata = [];
for ch = 1:size(data,1);
    % get snippets of good data
    autosnips = struct2cell(regionprops(data(ch,:)~=0,...
        data(ch,:), 'PixelValues'));
    [elength] = cellfun('size', autosnips,2) ;
    k=1;
    concatchn = [];
    % concatenate the good data
    while k <= length(elength);
            concatchn = [concatchn autosnips{k}];
            k = k+1;
    end
    concatdata = [concatdata; concatchn];
end
concatdata = [concatdata; zeros(1,size(concatdata,2))]; % set marker channel

% indicate process status
set(handles.text_process,'String','auto detect: done.');

% set display to clean data
set(handles.popupmenu_view,'Value',2);

% refresh GUI
updateGUI(handles);

% --- Executes on button press in pushbutton_reject.
function pushbutton_reject_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_reject (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global concatdata rawdata cleandata resetdata artdata

rawdata = concatdata; % replace clean data with data without artifacts
cleandata = rawdata; % replace clean data with data without artifacts
resetdata = rawdata;

artdata = nan(size(cleandata)); % reset detected artifacts

% refresh GUI
updateGUI(handles);

% --- Executes on button press in pushbutton_runICA.
function pushbutton_runICA_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_runICA (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
global cleandata Fs icaEEG wicaEEG meanvar comp icaComp

% get clean data without markers
data = cleandata(1:14,:);

% run wavelet enhanced ICA
icaMethod = get(handles.popupmenu_base,'Value'); % get ICA method

% run wavelet enhanced ICA
icaAlg = get(handles.popupmenu_algorithm,'Value'); % get ICA algorithm

% indicate method and status of process
nameMethod = {'status: ICA ','status: wICA '};
set(handles.text_process,'String', [nameMethod{icaMethod} 'processing...']);

% run ICA
[artRemEEG,icaEEG,wicaEEG,meanvar,comp,icaComp] = ...
    runICA(data,Fs,icaMethod,icaAlg);

% indicate end of process
set(handles.text_process,'String',[nameMethod{icaMethod} 'done.']);

% update clean data
cleandata(1:14,:) = artRemEEG;

% set display to clean data
set(handles.popupmenu_view,'Value',2);

% refresh GUI
updateGUI(handles);

% ======================================================================
% POPUPMENUES
% ======================================================================

% --- Executes on selection change in popupmenu_scale.
function popupmenu_scale_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu_scale (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu_scale contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu_scale

% refresh GUI
updateGUI(handles);

% --- Executes during object creation, after setting all properties.
function popupmenu_scale_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu_scale (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu_view.
function popupmenu_view_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu_view (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu_view contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu_view

% refresh GUI
updateGUI(handles);

% --- Executes during object creation, after setting all properties.
function popupmenu_view_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu_view (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu_base.
function popupmenu_base_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu_base (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu_base contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu_base


% --- Executes during object creation, after setting all properties.
function popupmenu_base_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu_base (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), ...
        get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on selection change in popupmenu_algorithm.
```

```
function popupmenu_algorithm_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu_algorithm (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu_algorithm contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu_algorithm


% --- Executes during object creation, after setting all properties.
function popupmenu_algorithm_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu_algorithm (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% ========================================================================
% EDIT TEXTS
% ========================================================================

function edit_length_Callback(hObject, eventdata, handles)
% hObject    handle to edit_length (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_length as text
%        str2double(get(hObject,'String')) returns contents of edit_length as a double
global Fs

% get epoch length from edit
eLength = floor(str2double(get(handles.edit_length,'String'))*Fs);
% set corresponding slider to value
set(handles.slider_epoch,'Value',eLength);
% refresh GUI
updateGUI(handles);

% --- Executes during object creation, after setting all properties.
function edit_length_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_length (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit_id_Callback(hObject, eventdata, handles)
% hObject    handle to edit_id (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_id as text
%        str2double(get(hObject,'String')) returns contents of edit_id as a double


% --- Executes during object creation, after setting all properties.
function edit_id_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_id (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% =========================================================================
% TEXT
% =========================================================================

% --- Executes during object creation, after setting all properties.
function text_file_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text_file (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% =========================================================================
% INTERACTIVE FEATURES
% =========================================================================

% --- Executes on mouse press over axes background.
function axes_display_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to axes_display (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global x

% get coordinates from mouse position
[x,y,button] = ginput(2);

% mark selection on data
```

```
selection = patch([x(1) x(2) x(2) x(1)],[0 0 1600 1600],'r','EdgeColor','none');
alpha(selection,0.2); % adjust transparency

% ======================================================================
% FUNCTIONS
% ======================================================================

% function for initialize all GUI components
% ---------------------------------------------------------------------
function initializeGUI(handles)

global Fs rawdata  cleandata vidLoaded  artdata wlength

% initialize variables
Fs = 128;       % sampling frequency of 128 Hz
wlength = 384;  % window length of 3 seconds (3*128 = 384 samples)
vidLoaded = 0;  % reset flag for video loaded

rawdata = zeros(15,100*Fs);     % initialize rawdata
cleandata = zeros(15,100*Fs);   % initialize cleandata
artdata = nan(15,100*Fs);       % reset dected artifacts

% initialize GUI components
set(handles.slider_display,'Value',1);      % set slider to fist data sample
set(handles.slider_epoch,'Value',10*Fs);    % show epoch of 10 sec
set(handles.popupmenu_scale,'Value',8)      % set scale to 100
set(handles.popupmenu_view,'Value',1)       % show raw data
set(handles.popupmenu_algorithm,'Value',1)       % show raw data
set(handles.popupmenu_base,'Value',2)       % show raw data
set(handles.axes_video, 'xtick',[],'ytick',[]);   % remove ticks from video axes
set(handles.axes_video, 'xticklabel',[],'yticklabel',[]); % remove labels from video axes

% update entire GUI
updateGUI(handles);

% function for updating all GUI components
% ---------------------------------------------------------------------
function updateGUI(handles)

global rawdata cleandata icaComp Fs vidFile vidLoaded videoIdx wicaEEG ...
    meanvar comp  artdata

% get length of data
xmax = size(cleandata,2);

% set slider maxima
set(handles.slider_display,'Max',xmax);
set(handles.slider_epoch,'Max',xmax);

% get epoch length
eLength = floor(get(handles.slider_epoch,'Value'));

% prevent exceeding indexes
if eLength < 0.5        % set epochs shorter than 0.5s to 0.5s
```

```
    eLength = 0.5;
end
if(eLength>xmax)
    eLength = xmax;
    set(handles.slider_epoch,'Value',xmax);
end
set(handles.text_eLength,'String',sprintf('%.1f sec',eLength/Fs));
set(handles.edit_length,'String',sprintf('%.1f', eLength/Fs));

% get epoch start in samples
smpStart = round(get(handles.slider_display,'Value'));

% prevent exceeding index
if(smpStart < 1)
    smpStart = 1;
end
if(smpStart>xmax)
    smpStart = xmax-eLength-1;
end
% set slider to epoch start
set(handles.slider_display,'Value',smpStart);

% update textfield for time
set(handles.text_time,'String',sprintf('Total time: %.2f sec',xmax/Fs));

% get scaling parameter
contents = cellstr(get(handles.popupmenu_scale,'String'));
scale = str2num(contents{get(handles.popupmenu_scale,'Value')});

% get display mode for view
mode = get(handles.popupmenu_view,'Value');

switch(mode)
    case 1 % rawdata
        cla reset
        data = rawdata; % get rawdata

        % extract epoch from data
        epoch = extractEpoch(data,smpStart,eLength);
        % plot epoch
        ChanTitles = {'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz' 'Mkr'};
        Title = 'Raw EEG Data';
        set(handles.axes_display,'HitTest','off');
        PlotEEG(handles,epoch,Fs,smpStart,scale,ChanTitles,Title,'k');
        % reactivate mouse selection
        set(handles.axes_display, 'ButtonDownFcn', ...
            {@axes_display_ButtonDownFcn,handles});
    case 2 % clean data
        cla reset
        data = cleandata; % get clean data
        art = artdata;    % get detected artifacts
        % extract epoch from data
        epoch = extractEpoch(data,smpStart,eLength);
        epochart = extractEpoch(art,smpStart,eLength);

        % plot epoch
        ChanTitles = {'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz' 'Mkr'};
        Title = 'Processed EEG Data';
        set(handles.axes_display,'HitTest','off');
        % plot data
        PlotEEG(handles,epoch,Fs,smpStart,scale,ChanTitles,Title,'k');
        hold on
        % mark detected artifacts
        PlotEEG(handles,epochart,Fs,smpStart,scale,ChanTitles,Title,'r');
        hold off

        % reactivate mouse selection
        set(handles.axes_display, 'ButtonDownFcn', ...
            {@axes_display_ButtonDownFcn,handles});
    case 3 % ICA components
        cla reset
        [N,M] = size(icaComp);  % get number of components
        dispComp = zeros(N+1,M); % initialize displayed components (no markers)
        dispComp(1:N,:) = icaComp;  % get ica components
        dispComp(end,:) = rawdata(end,:); % insert markers
        % extract epoch from compoonents
        data = dispComp;
        epoch = extractEpoch(data,smpStart,eLength);
        % plot epoch
        ChanTitles = {'1' '2' '3' '4' '5' '6' '7' ...
            '8' '9' '10' '11' '12' '13' '14' 'Mkr'};
        Title = 'Independent Components (ICA)';
        set(handles.axes_display,'HitTest','off');
        PlotEEG(handles,epoch,Fs,smpStart,scale,ChanTitles,Title,'k');
        set(handles.axes_display, 'ButtonDownFcn', ...
            {@axes_display_ButtonDownFcn,handles});
    case 4 % wavelet based ICA components
        cla reset
        [N,M] = size(wicaEEG);
        dispComp = zeros(size(rawdata)); % initialize displayed components
        dispComp(1:N,:) = wicaEEG;       % get wavelet based ica components
        dispComp(end,:) = rawdata(end,1:M); % get markers
        % extract epoch from data
        data = dispComp;
        epoch = extractEpoch(data,smpStart,eLength);
        % plot epoch
        ChanTitles = {'1' '2' '3' '4' '5' '6' '7' ...
            '8' '9' '10' '11' '12' '13' '14' 'Mkr'};
        Title = 'Independent Components (wICA)';
        set(handles.axes_display,'HitTest','off');
        PlotEEG(handles,epoch,Fs,smpStart,scale,ChanTitles,Title,'k');
        set(handles.axes_display, 'ButtonDownFcn', ...
            {@axes_display_ButtonDownFcn,handles});
    case 5 % screen plot
        cla reset
        plot([1:length(meanvar)],meanvar,'b'),hold on
        plot([1:length(comp)],meanvar(comp),'ok');
        title('SVD')
        xlabel('components')
        ylabel('singular values')
end

% show video if available
if (vidLoaded)
    % calculate frame for video display
    vidFrame = floor(smpStart/Fs*vidFile.FrameRate)+1;
    showVideo(handles,vidFile,vidFrame,videoIdx);
end

% function for plotting EEG data
% ------------------------------------------------------------------------
 function PlotEEG(handles,data,Fs,start,Scale,ChanTitles,Title,col)

% make display current axes
axes(handles.axes_display);

% get number of channels and samples of data
[nChs,nPts] = size(data);

% generate time vector
t = (start:start+nPts-1)/Fs;

% initialize variables
Ct = {};
tt = {};
dt = floor(nPts/10); % define interval of time labels

% define time lables
for tlabel = ((start:dt:start+nPts)/Fs)
    tt = [tt; {sprintf('%.1f',tlabel)}];
end

% plot EEG channels
for k = 1:nChs-1,
    plot(t, real((nChs - k)*Scale + data(k,:)),col);
    hold on
    if isempty(ChanTitles),
        Ct = [Ct; {num2str(nChs - k + 1)}];
    else
        Ct = [Ct; ChanTitles(nChs - k + 1)];
    end
end

% plot markers
Marker = data(15,:);    % get marker data from 15th channel
Marker(Marker>0) = 1500;
mrkIdx = find(Marker>0); % find marker in marker data

% plot marker on GUI
stem(t,Marker,'r','Linewidth',2,'Marker','none');
```

```
Ct = [Ct; ChanTitles(1)];
b = num2str(round(data(nChs,mrkIdx))); c = cellstr(b);
dx = 0.1; dy = 0.1; % displacement so the text does not overlay the data point
% label the marker
text(t(mrkIdx)+dx, Marker(mrkIdx)+dy, c,'color','r','FontWeight','bold')

% adjust axes
hold off;
xlim([start start+nPts]/Fs);
ylim([-1 nChs+1]*Scale);
set(gca,'XTick',(start:dt:start+nPts)/Fs);
set(gca,'XTickLabel',tt);
set(gca,'YTick', (0:1:nChs-1)*Scale);
set(gca,'YTickLabel',Ct);

% set title and x-label
title(Title);
xlabel('Time (sec)')

% function to show video
% ----------------------------------------------------------------------
function showVideo(handles,vidfile,vidFrame,videoIdx)

% get frame index
frameIdx = videoIdx(vidFrame);
% read video file at frame index
vid = read(vidfile,frameIdx);
% display frame
image(vid,'parent',handles.axes_video);
% remove axes for video
set(handles.axes_video, 'xtick',[],'ytick',[]);
set(handles.axes_video, 'xticklabel',[],'yticklabel',[]);

% function to extract displayed epoch from data stream
% ----------------------------------------------------------------------
function epoch = extractEpoch(data,start,length)

% prevent over/underflow
if ((start+length-1) > size(data,2))
    stop = size(data,2);
else
    stop = start+length-1;
end
% extract data from start to stop
epoch = data(:,start:stop);

% function to re-reference data to average
% ----------------------------------------------------------------------
function avgRefEEG = avgReference(data)

avgCh = mean(data); % calculate average over channels
for ch = 1:14
    avgRefEEG(ch,:) = data(ch,:) - avgCh; % remove average value from data
end
```

```
% function to remove artifacts using ICA (normal and wavelet based
% ------------------------------------------------------------------------
function [artRemEEG,icaEEG,wicaEEG,meanvar,comp,icaComp] = ...
        runICA(data,Fs,icaMethod,icaAlg)

% initialize variables
icaEEG = [];
wicaEEG = [];
icaComp = [];
comp = [];
meanvar = [];



% obtain unmixing matrix

switch(icaAlg)
    case 1
        [weight, sphere] = runica(data); % run InfoMax algorithm
        W = weight*sphere;    % obtain unmixing matrix
        Winv = inv(W); % obtain mixing matrix
        icaEEG = W*data; % compute components

        % set thrshold for wavelet artifact detection
        threshold = 1.25;

        % clean components with thresholding of wavelets
        [wicaEEG, opt]= RemoveStrongArtifacts(icaEEG, 1:size(data,1),...
            threshold, Fs);

    case 2
        W = jader(data); % run JADE algorithm
        Winv = inv(W); % obtain mixing matrix
        icaEEG = W*data; % compute components

        % set thrshold for wavelet artifact detection
        threshold = 1.25;

        % clean components with thresholding of wavelets
        [wicaEEG, opt]= RemoveStrongArtifacts(icaEEG, 1:size(data,1),...
            threshold, Fs);

    case 3
        [Winv,icaEEG] = sobi(data); % run SOBI algorithm

        % set wicaEEG variable to ica EEG (wavelet based removal not
        % applicable)
        wicaEEG = icaEEG;
end


% compute variances without backprojecting to save time and memory -sm 7/05
[chans,frames] = size(data);
```

```
meanvar = sum(Winv.^2).*sum((data').^2)/((chans*frames)-1); % from Rey Ramirez 8/07

% sort variances from large to small
[sortvar, windex] = sort(meanvar);
windex = windex(chans:-1:1); % order large to small
meanvar = meanvar(windex);

% calculate number of components that contribute less than 95% to variance
var95 = 0.95*sum(meanvar);
svsum = meanvar(1);
oldsvsum = 1;
k = 1;
while (svsum<var95) && (k<length(meanvar) && (svsum-oldsvsum)/oldsvsum>0.2)
    k=k+1;
    oldsvsum = svsum;
    svsum = sum(meanvar(1:k)); % sum up variances
end
comp = 1:k; % save "artifact" components

% save sorted components
icaComp = icaEEG(windex,:);

% set "artifact" components to zero
icaEEG(windex(1:k),:) = 0;

% remix components to eeg channels using mixing matrix
if icaMethod == 1
    artRemEEG = real(Winv*icaEEG);
elseif icaMethod == 2
    artRemEEG = real(Winv*wicaEEG);
end

% function to plot spectrum (PSD)
% -----------------------------------------------------------------------
function plotPSD(spec,f)

hold off
% define limits of plot
xlim = [0 40];
ylim = [0 1.1*max(spec(:))];

% get number of channels
num = size(spec,1);

% define color vector
cc = hsv(num);

% plot spectra with changing colors
for i=1:num
    plot(f,spec(i,:),'color',cc(i,:)),hold on
end
axis([xlim ylim])
xlabel('Frequency [Hz]')
ylabel('Spectral Density (Hz^{-1})');
```

```
% adjust labels on Y axis
if num < 14
    legendnr = {'1','2','3','4','5','6','7','8','9','10','11','12','13'};
    legend(legendnr(1:num),'fontsize',8);
else
    legend({'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8);
end
```

4.1.1 Function: selectData(…)

```
% Grand Valley State University, Master of Science in Engineering
% ========================================================================
% author: Nadina Zweifel
% date: 3/16/2016
% advisor: Dr. Samhita Rhodes
% ------------------------------------------------------------------------
% function: data = selectData(data,scale,Fs)
% this function reads the user's mouse cursor position when a selection in
% the data is made.
% input: data, scale, Fs
% data is a matrix with the time series where the rows are the channels and
% the columns the samples. scale is the vertical scaling factor of the
% displayed data plot. Fs is the sampling frequency of the data.
% output: data
% data returns the indexes for the selected region.
% ------------------------------------------------------------------------

function data = selectData(data,scale,Fs)

% get the mouse cursor position from the user
[x,y] = ginput(2);
% mark the user's selection in the data with a colored rectangle
selection = patch([x(1) x(2) x(2) x(1)],[0 0 15 15]*scale,'r','EdgeColor','none');
% set the rectangle transparent
alpha(selection,0.2);
% convert the curser position to samples
xSmpl = round(x*Fs);
% determin left and right boundery
xleft = min(xSmpl);
xright = max(xSmpl);

% return the indexes of the selected regions preventing that the indexes
% exceed the dimensions
if xright>size(data,2)
    data = data(:,xleft:end);
elseif xleft<1
    data = data(:,1:xright);
else
    data = data(:,xleft:xright);
```

```
end
end
```

4.1.2 Function: ploteeg(…)

```
% Grand Valley State University, Master of Science in Engineering
% ========================================================================
% author: Nadina Zweifel
% date: 3/16/2016
% advisor: Dr. Samhita Rhodes
% ------------------------------------------------------------------------
% function: ploteeg(data,Fs,start,scale,chanTitles,title,col)
% function to plot the time series of EEG data. The function is based on
% the code for wavelet enhanced ICA by Castellanos & Makarov 2006
% input: data, Fs, start, scale, chanTitles, title, col
% data is a matrix with the time series where the rows are the channels and
% the columns the samples using the sampling frequency Fs. start is the
% index of the first sample displayed in the figure. scale defines the
% vertical scaling factor of the EEG data. chanTitles consists of cell
% array with strings for labeling the channels on the vertical axis. col
% contains the values for defining the color of the plot.
% ------------------------------------------------------------------------


function ploteeg(data,Fs,start,scale,chanTitles,title,col)

% check if at least data variable is an input
if nargin < 1,
    disp('No arguments');
    help PlotEEG
    return;
end
% get dimensions of the data - nChns: # channels, nPts: # samples
[nChs,nPts] = size(data);

% check number of channels, if too high it's assumed to be time samples
if nChs > 20,
    disp('Too many channels (try to transpose the data)');
    return
end

% define time vector
t = (start:start+nPts-1)/Fs;

% initialize variables
Ct = {};
tt = {};

% define time resolution on plot
dt = floor(nPts/10);
```

```
% define lables for time axis
for tlabel = ((start:dt:start+nPts)/Fs)
tt = [tt; {sprintf('%.1f',tlabel)}];
end

% plot the data in the correct scaling defined by variable scale
for k = 1:nChs,
    plot(t, real((nChs - k)*scale + data(k,:)),col);
    hold on
    % add labels for channels
    if isempty(chanTitles),
        Ct = [Ct; {num2str(nChs - k + 1)}];
    else
        Ct = [Ct; chanTitles(nChs - k + 1)];
    end
end

% add marker label to last row of data
Ct = [Ct; chanTitles(1)];

hold off;

% define plot limits
xlim([start start+nPts]/Fs);
ylim([-1 nChs+1]*scale);
% add title to plot
title(title);

% set time labels
set(gca,'XTick',(start:dt:start+nPts)/Fs);
set(gca,'XTickLabel',tt);
% set channels labels
set(gca,'YTick', (0:1:nChs-1)*scale);
set(gca,'YTickLabel',Ct);
% add label for x-axis
xlabel('Time (sec)')
```

4.1.3 Function: wICA(…)

```
function Data_wICA = wICA(Data,Fs)
% For wICA the algorithm of Valeri A. Makarov was used:
% This code is for illustration of the method described in:
% N.P. Castellanos, and V.A. Makarov (2006). "Recovering EEG brain signals: Artifact
% suppression with wavelet enhanced independent component analysis"
% J. Neurosci. Methods, 158, 300–312.
%
% Requirements: runica from EEGLAB toolbox, and rwt - Rice Wavelet Toolbox
% (both freely available in internet).
%
% This code is copyright © by the authors, and we hope you acknowledge our
% work. We distribute it in the hope that it will be useful, but without any warranty.
```

```
%
% Author: Valeri A. Makarov
% e-mail: vmakarov@opt.ucm.es
%
% 2006
%
% Find independent components
% EEGLAB is required!!! You can also use other algorithms, e.g. fICA.
% Note, the use of long (in time) data sets reduces the algorithm performance
% see for details the abovementioned paper.

Fnyq  = Fs/2;
% F_notch = 50; % Notch at 50 Hz
% [b,a] = iirnotch(F_notch/Fnyq, F_notch/Fnyq/20);
% Data  = filtfilt(b,a, Data);

% Conventional High Pass Filter
% This is an optional step (suppress low <4Hz frequency noise)
F_cut = 4;
[b,a] = ellip(1, 0.5, 20, F_cut/Fnyq, 'high');
Data  = filtfilt(b,a, Data);

% Remove mean values from the channels and plot raw data
Data  = detrend(Data,'constant');
% Transpose the data matrix to get (channel x time) orientation
Data = Data';

% wICA

[weight, sphere] = runica(Data, 'verbose', 'off');

W = weight*sphere;    % EEGLAB --> W unmixing matrix
icaEEG = W*Data;      % EEGLAB --> U = W.X activations

[icaEEG2, opt]= RemoveStrongArtifacts(icaEEG, (1:14), 1.25, Fs);
Data_wICA = inv(W)*icaEEG2;

end
```

4.1.4 Function: RemoveStrongArtifact(…)

```
function [icaEEG, opt] = RemoveStrongArtifacts(icaEEG, Comp, Kthr, F)
%
% This function denoise high amplitude artifacts (e.g. ocular) and remove them from the
% Independent Components (ICs).
%
% INPUT:
%
% icaEEG - matrix of ICA components (Nchanel x Nobservations)
%
% Comp   - # of ICs to be denoised and cleaned (can be a vector)
```

```
%
% Kthr   - threshold (multiplayer) for denoising of artifacts
%          (default Kthr = 1.15)
%
% F      - acquisition frequency
%          (default F = 256 Hz)
%
% OUTPUT:
%
% icaEEG - matrix of cleaned indenpendent components
%
% opt    - vector of threshold values used for filtering of corresponding
%          ICs
%
% NOTE: If a component has no artifacts of a relatively high amplitude
%       the function will skip this component (no action), dispaly a
%       warning and the corresponding output "opt" will be set to zero.
%
% Valeri A. Makarov, vmakarov@opt.ucm.es
% ver 0.1 Sept. 2005
% ver 0.2 May   2006
%

if nargin < 2,
    disp('At least two arguments are required!');
    help RemoveStrongArtifacts
    return;
end
if nargin < 3 || isempty(Kthr), Kthr = 1.15; end
if nargin < 4 || isempty(F), F = 256; end
L = round(F*0.1);
[Nchan, Nobser] = size(icaEEG);
if Nchan > Nobser,
    error('Problem with data orientation, try to transpose the matrix!');
end
N = 2^floor(log2(Nobser));
h = daubcqf(6);
opt = zeros(1,length(Comp));
for c=1:length(Comp),
    Y = icaEEG(Comp(c),1:N);     % cth component from 1 to N
    Sig = median(abs(Y)/0.6745); % get measure of data amplitude
    Thr = 4*Sig;                 % determine threshold
    idx = find(abs(Y) > Thr);    % find indices above threshold
    idx_ext = zeros(1,length(idx)*(2*L+1)); % extend artifact part
    for k=1:length(idx),
        idx_ext((2*L+1)*(k-1)+1:(2*L+1)*k) = [idx(k)-L:idx(k)+L];
    end
    id_noise=setdiff((1:N), idx_ext);   % indices of signal of interest
    id_artef=setdiff((1:N), id_noise);  % indices of artifacts
    if isempty(id_artef),
        %disp(['The component #' num2str(Comp(c)) ' has passed unchanged']);
        continue;
    end
    thld = 3.6;     % threshold for artifact removal
    KK = 100;
    LL = floor(log2(length(Y)));
    [xl, xh] = mrdwt(Y, h, LL); % wavelet transform (without downsampling)
    while KK > Kthr,            % test vs. input parameter Kthr = 1.25
        thld = thld + 0.5;      % adjust treshold
        xh = HardTh(xh, thld);  % hard thresholding of detail coefficients
%         xl = HardTh(xl, thld);
        xd = mirdwt(xl,xh,h,LL); % backprojection of artifacts with inverse wt
        xn = Y - xd;             % subtract artifacts
        cn=corrcoef(Y(id_noise),xn(id_noise)); % calculate ratios
        cd=corrcoef(Y(id_noise),xd(id_noise));
        ca=corrcoef(Y(id_artef),xd(id_artef));

        KK = ca(1,2)/cn(1,2);
        KKnew = ca(1,2)/cd(1,2);
    end
%     disp(['KK(ca/cn): ' num2str(KK)])
%     disp(['KK(ca/cd): ' num2str(KKnew)])
    opt(c) = thld;          % save optimal threshold
    Y = icaEEG(Comp(c),end-N+1:end);  % apply wavelet thresholding to rest of data
    icaEEG(Comp(c),1:N) = xn;
    LL = floor(log2(length(Y)));
    [xl, xh] = mrdwt(Y, h, LL);
    xh = HardTh(xh, thld);
    xd = mirdwt(xl,xh,h,LL);
    xn = Y - xd;
    icaEEG(Comp(c),N+1:end) = xn(end-(Nobser-N)+1:end);
%     disp(['The component #' num2str(Comp(c)) ' has been filtered']);
end
```

4.2 Code: Theta Rhythm

4.2.1 Main Script: thetaRhythm.m

```
% Grand Valley State University, Master of Science in Engineering
% =========================================================================
% author: Nadina Zweifel
% date: 3/16/2016
% advisor: Dr. Samhita Rhodes
%
% modified by: Joshua Usoro
% date: 04/03/2017
% advisor: Dr. Samhita Rhodes
% -------------------------------------------------------------------------
% Master Thesis: Main script for finding the theta rhythm of the subject
% -------------------------------------------------------------------------

clear all, clc, close all

% load data from a file
%load('P7newdata.mat')
```

```
% define the start of each phase
% weeks = [1,5,10,15,20]; %p5
% weeks = [1,6,11,16,21]; %p6
weeks = [1,4,9,12,17]; %p7

% define frequency range limits
lf = 2;
hf = 12;

% calculate sample points for limits
lfsmp = floor(lf*4);
hfsmp = ceil(hf*4);

% define frequency subband width
wsub = 4;

% get number of sessions
ntrials = size(data_rest1,2);

% define normalization method: norm=1 normalizes spectrum with overall mean
% power across channels and frequencies
norm = 1;

% compute spectra and sub the frequencies across wsub (=4) points
for trial = 1:ntrials
    % resting 1
    [c_psd_rest1{trial},freq,psd_subs,err_subs,freq_subs] = ...
        psdsubswitherr(data_rest1{trial},lf,hf,wsub,norm);
    c_psd_subs_rest1{trial} = psd_subs;
    c_err_subs_rest1{trial} = err_subs;
    % interaction
    [c_psd_interact{trial},freq,psd_subs,err_subs,freq_subs] = ...
        psdsubswitherr(data_interact{trial},lf,hf,wsub,norm);
    c_psd_subs_interact{trial} = psd_subs;
    c_err_subs_interact{trial} = err_subs;
end

% transform the cell arrays to matrices
psd_subs_rest1 = (cat(3,c_psd_subs_rest1{:}));
psd_subs_interact = (cat(3,c_psd_subs_interact{:}));
err_subs_rest1 = (cat(3,c_err_subs_rest1{:}));
err_subs_interact = (cat(3,c_err_subs_interact{:}));

% calculate the percentage change from resting 1 to interaction
psd_change = 100*(psd_subs_interact-psd_subs_rest1)./psd_subs_rest1;

% remove 1st trial due to artifactual data
trials_a1 = [1 2 3 4];

% calculate mean for first baseline phase over the 3 points
mean_a1_interact = mean(psd_subs_interact(:,:,trials_a1),3);
mean_a1_rest1 = mean(psd_subs_rest1(:,:,trials_a1),3);
```

```
% calculate the percentage change of the means
mean_a1_change = 100*(mean_a1_interact-mean_a1_rest1)./mean_a1_rest1;

% test the change for significance using one standard error
[psd_sig,psd_sig_mag,psd_mean,psd_change] = ...
    test22SDtest(psd_subs_rest1,psd_subs_interact,err_subs_interact,weeks);
```

display results

```
% define limits of colorbar
zmin = 0;
zmax = 10;
% define resolution colormap: continuous -> ncol=0
ncol = 0;
% define colorbar label
cb_name = 'PSD (Vm^2)';
% define frequency sub vector
freq_subs = (lfsmp:wsub:hfsmp)/4;

% get number of channels and subs
[nchannels,nsubs] = size(mean_a1_rest1(:,:,1));

% resting 1
figure
imagesc(mean_a1_rest1)
setColorbar(zmin,zmax,cb_name,0,ncol);
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',10)
title('Spectral mean power A1 (no interaction 1)')
xlabel('Frequency sub-bands (Hz)')
ylabel('Channels')
pbaspect([1 1 1])

% interaction
figure
imagesc(mean_a1_interact)
setColorbar(zmin,zmax,cb_name,0,ncol);
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',10)
title('Spectral mean power A1 (interaction)')
xlabel('Frequency sub-bands (Hz)')
ylabel('Channels')
pbaspect([1 1 1])
```

display results for percentage change

```
zmin = -70;
zmax = 70;
% define colorbar label
cb_name = 'change (%)';
% define resolution colormap: continuous -> ncol=0
ncol = 0;

% display figure
figure
imagesc(mean_a1_change)
setColorbar(zmin,zmax,cb_name,1,ncol);
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',10)
title('Change PSD: A1(no interaction 1) -> A1(interaction)')
xlabel('Frequency sub-bands (Hz)')
ylabel('Channels')
pbaspect([1 1 1])

% display results for significant percentage change
figure
imagesc(psd_sig_mag(:,:,1))
setColorbar(zmin,zmax,cb_name,1,ncol);
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',10)
title('Significant change (no interaction 1/interact) of mean PSD in 1st baseline (A1)')
xlabel('Frequency sub-bands (Hz)')
ylabel('Channels')
pbaspect([1 1 1])
```

4.3 Code: Spectral Analysis

4.3.1 Main Script: spectralAnalysis.m

```
% Grand Valley State University, Master of Science in Engineering
% =========================================================================
% author: Nadina Zweifel
% date: 3/16/2016
% advisor: Dr. Samhita Rhodes
%
% modified by: Joshua Usoro
% date: 04/03/2017
% advisor: Dr. Samhita Rhodes
% ------------------------------------------------------------------------
% Master Thesis: Main script for spectral analysis of EEG data. The script
% loads the files and transforms the time series to frequency domain for
% spectral analysis.
% ------------------------------------------------------------------------

clear all, clc, close all

% load data from a file
%load('P7newdata.mat')

% define the start of each phase
% weeks = [1,5,10,15,20]; %p5
% weeks = [1,6,11,16,21]; %p6
weeks = [1,4,9,12,17]; %p7

% define frequency range limits
lf = 2;
hf = 12;

% calculate sample points for limits
lfsmp = floor(lf*4);
hfsmp = ceil(hf*4);

% define frequency sub width
wsub = 4;

% get number of sessions
ntrials = size(data_rest1,2);

% define normalization method: norm=1 normalizes spectrum with overall mean
% power across channels and frequencies
norm = 1;

% compute spectra and sub the frequencies across wsub (=4) points
for trial = 1:ntrials
    % resting 1
    [c_psd_rest1{trial},freq,psd_subs,err_subs,freq_subs] = ...
        psdsubswitherr(data_rest1{trial},lf,hf,wsub,norm);
    c_psd_subs_rest1{trial} = psd_subs;
    c_err_subs_rest1{trial} = err_subs;
    % resting 2
    [c_psd_rest2{trial},freq,psd_subs,err_subs,freq_subs] = ...
        psdsubswitherr(data_rest2{trial},lf,hf,wsub,norm);
    c_psd_subs_rest2{trial} = psd_subs;
    c_err_subs_rest2{trial} = err_subs;
    % interaction
    [c_psd_interact{trial},freq,psd_subs,err_subs,freq_subs] = ...
        psdsubswitherr(data_interact{trial},lf,hf,wsub,norm);
    c_psd_subs_interact{trial} = psd_subs;
    c_err_subs_interact{trial} = err_subs;
end
```

```
% transform the cell arrays to matrices
psd_rest1 = (cat(3,c_psd_rest1{:}));
psd_rest2 = (cat(3,c_psd_rest2{:}));
psd_interact = (cat(3,c_psd_interact{:}));

psd_subs_rest1 = (cat(3,c_psd_subs_rest1{:}));
psd_subs_rest2 = (cat(3,c_psd_subs_rest2{:}));
psd_subs_interact = (cat(3,c_psd_subs_interact{:}));

err_subs_rest1 = (cat(3,c_err_subs_rest1{:}));
err_subs_rest2 = (cat(3,c_err_subs_rest2{:}));
err_subs_interact = (cat(3,c_err_subs_interact{:}));

% calculate the percentage change from resting 1 to interaction
psd_subs_change = (psd_subs_rest2 - psd_subs_rest1)./psd_subs_rest1*100;

% save the workspace to a file
%save('vars_6.mat')
```

display results

```
% define channel labels
chnlabels = {'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
             'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'};
```

resting 1

```
% apply two-standard deviation band method
[psd_sig,psd_sig_mag,psd_mean] = test2SDtest(psd_subs_rest1,err_subs_rest1,weeks);

% get number of channels and subs
[nchannels,nsubs] = size(psd_sig(:,:,1));

% display results with significant score

% define frequency vector
freq_subs = (lfsmp:wsub:hfsmp)/4;

% define margin around subplot
margin = 0.06;

% define paramter for colorbar
zmin = -4.5;
zmax = 4.5;
cb_name = 's-value';
% define resolution colormap: continuous -> ncol=0
ncol = 8;

figure
% display change A1-B1
h=subplot(131);
% adjust margins of subplot
```

```
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(psd_sig(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('PSD Significance: A1->B1 (no interaction 1)')
xlabel('Frequency sub-bands (Hz)')
ylabel('Channels')
pbaspect([1 1 1])

% display change A1-A2
h=subplot(132);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(psd_sig(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('PSD Significance: A1->A2 (no interaction 1)')
xlabel('Frequency sub-bands (Hz)')
%ylabel('Channels')
pbaspect([1 1 1])

% display change A1-B2
h=subplot(133);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(psd_sig(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
```

```
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('PSD Significance: A1->B2 (no interaction 1)')
xlabel('Frequency sub-bands (Hz)')
%ylabel('Channels')
pbaspect([1 1 1])

% display results with significant magnitude
% set colorbar properties
zmin = -300;
zmax = 300;
cb_name = 'Change (%)';
% define resolution colormap: continuous -> ncol=0
ncol = 0;

figure
% display results for A1-B1
h=subplot(131);
% adjust margin of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(psd_sig_mag(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change PSD: A1->B1 (no interaction 1)')
xlabel('Frequency sub-bands (Hz)')
ylabel('Channels')
pbaspect([1 1 1])

% display results for A1-A2
h=subplot(132);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% dislay result matrix
imagesc(psd_sig_mag(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change PSD: A1->A2 (no interaction 1)')
xlabel('Frequency sub-bands (Hz)')
%ylabel('Channels')
pbaspect([1 1 1])

% display results for A1-B2
h=subplot(133);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(psd_sig_mag(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change PSD: A1->B2 (no interaction 1)')
xlabel('Frequency sub-bands (Hz)')
%ylabel('Channels')
pbaspect([1 1 1])
```

display results for resting 2

```
% apply two-standard deviation band method
[psd_sig,psd_sig_mag,psd_mean] = test2SDtest(psd_subs_rest2,err_subs_rest2,weeks);

% set colorbar properties
zmin = -4.5;
zmax = 4.5;
cb_name = 's-value';
% define resolution colormap: continuous -> ncol=0
ncol = 8;

figure
% display results with significant score

% display results for A1-B1
h=subplot(131);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(psd_sig(:,:,2))
% set colorbar
```

```
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('PSD Significance: A1->B1 (no interaction 2)')
xlabel('Frequency sub-bands (Hz)')
ylabel('Channels')
pbaspect([1 1 1])

% display results for A1-A2
h=subplot(132);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(psd_sig(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('PSD Significance: A1->A2 (no interaction 2)')
xlabel('Frequency sub-bands (Hz)')
%ylabel('Channels')
pbaspect([1 1 1])

% display results for A2-B2
h=subplot(133);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(psd_sig(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('PSD Significance: A1->B2 (no interaction 2)')
xlabel('Frequency sub-bands (Hz)')
```

```
%ylabel('Channels')
pbaspect([1 1 1])

% display results for significant change (magnitude)
% set colorbar properties
zmin = -300;
zmax = 300;
cb_name = 'Change (%)';
% define resolution colormap: continuous -> ncol=0
ncol = 0;

figure
% display results for A1-B1
h=subplot(131);
% adjust margin for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(psd_sig_mag(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change PSD: A1->B1 (no interaction 2)')
xlabel('Frequency sub-bands (Hz)')
ylabel('Channels')
pbaspect([1 1 1])

% display results for A1-B2
h=subplot(132);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(psd_sig_mag(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change PSD: A1->A2 (no interaction 2)')
xlabel('Frequency sub-bands (Hz)')
%ylabel('Channels')
```

```
pbaspect([1 1 1])

% display results for A2-B2
h=subplot(133);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(psd_sig_mag(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change PSD: A1->B2 (no interaction 2)')
xlabel('Frequency sub-bands (Hz)')
%ylabel('Channels')
pbaspect([1 1 1])
```

display results for rest1/resting 2

```
% apply two-standard deviation band method to detect change from rest1- to
% resting 2
[psd_sig,psd_sig_mag,psd_mean,psd_change] = ...
    test22SDtest(psd_subs_rest1,psd_subs_rest2,err_subs_rest2,weeks);

% set margin parameter
margin = 0.04;

% display results for significance
% set colorbar parameters
zmin = -4.5;
zmax = 4.5;
cb_name = 's-value';
% define resolution colormap: continuous -> ncol=0
ncol = 8;

figure
% display results for A1(rest1) - A2(rest2)
h=subplot(221);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(psd_sig(:,:,1))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
```

```
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('PSD Significance: A1(no interact 1)->A1(no interact 2)')
xlabel('Frequency sub-bands (Hz)')
ylabel('Channels')
pbaspect([1 1 1])

% display results for B1(rest1) - B1(rest2)
h=subplot(222);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(psd_sig(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('PSD Significance: B1(no interact 1)->B1(no interact 2)')
xlabel('Frequency sub-bands (Hz)')
%ylabel('Channels')
pbaspect([1 1 1])

% display results for A2(rest1) - A2(rest2)
h=subplot(223);
% set margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(psd_sig(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('PSD Significance: A2(no interact 1)->A2(no interact 2)')
xlabel('Frequency sub-bands (Hz)')
ylabel('Channels')
pbaspect([1 1 1])
```

```
% display results for B2(rest1) - B2(rest2)
h=subplot(224);
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
imagesc(psd_sig(:,:,4))
setColorbar(zmin,zmax,cb_name,1,ncol);
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('PSD Significance: B2(no interact 1)->B2(no interact 2)')
xlabel('Frequency sub-bands (Hz)')
%ylabel('Channels')
pbaspect([1 1 1])

% display significant change in magnitude
% adjust colorbar parameters
zmin = -300;
zmax = 300;
cb_name = 'Change (%)';
% define resolution colormap: continuous -> ncol=0
ncol = 0;

figure
% display results for A1(rest1) - A1(rest2)
h=subplot(221);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(psd_sig_mag(:,:,1))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change PSD: A1(no interact 1)->A1(no interact 2)')
xlabel('Frequency sub-bands (Hz)')
ylabel('Channels')
pbaspect([1 1 1])

% display results for B1(rest1) - B1(rest2)
h=subplot(222);
% adjust margins for subplot
p = get(h, 'pos');
```

```
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display results matrix
imagesc(psd_sig_mag(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change PSD: B1(no interact 1)->B1(no interact 2)')
xlabel('Frequency sub-bands (Hz)')
%ylabel('Channels')
pbaspect([1 1 1])

% display results for A2(rest1) - A2(rest2)
h=subplot(223);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display results matrix
imagesc(psd_sig_mag(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change PSD: A2(no interact 1)->A2(no interact 2)')
xlabel('Frequency sub-bands (Hz)')
ylabel('Channels')
pbaspect([1 1 1])

% display results for B2(rest1) -> B2(rest2)
h=subplot(224);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(psd_sig_mag(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca,'XTick',1:nsubs)
xticklabel = cellstr(num2str(freq_subs',2));
set(gca,'XTickLabel',xticklabel,'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change PSD: B2(no interact 1)->B2(no interact 2)')
xlabel('Frequency sub-bands (Hz)')
%ylabel('Channels')
pbaspect([1 1 1])
```

4.3.2 Function: test2SD(…)

```
% Grand Valley State University, Master of Science in Engineering
% ========================================================================
% author: Nadina Zweifel
% date: 3/16/2016
% advisor: Dr. Samhita Rhodes
%
% modified by: Joshua Usoro
% date: 04/03/2017
% advisor: Dr. Samhita Rhodes
% ------------------------------------------------------------------------
% function: [psd_sig,psd_sig_mag,psd_mean] = test2SD(psd,substd)
% this function applies the two-standard deviation method to the input
% data which contains data of different phases (baseline and intervention).
% The function assumes that the phases consists of 4 data points and
% consideres the first 4 points as the baseline. Significance is tested by
% counting the data points exceeding the 2*SE band (SE=standard error).
% Cases that exceed the band 2 times or more are considered as significant.
% input: psd,substd
% psd represents the power spectral density of the EEG data under
% consideration. The input data is subned by taking the average across
% data points in a particular frequency range. Hence, psd contains the
% means of the subs while substd contains the standard deviation of each
% sub.
% output: psd_sig, psd_sig_mag, psd_mean
% psd_sig is a matrix containing the significance score (s-value) for each
% sub and channel in respect of the baseline. Unsignificant changes are
% represented with a zero. The s-value is 2 or higher which is considered
% as significant. psd_sig_mag is of the same type as psdsig but instead of
% s-values it contains the percentage change score for signifcant changes.
% psd_mean returns just the mean value for each sub and channel for each
% phase.
% ------------------------------------------------------------------------

function [psd_sig,psd_sig_mag,psd_change] = test2SDtest(psd,substd,weeks)
    [nchannels,nsubs,nsessions] = size(psd);

    % calculate the baseline values in the first phase of the input data
    psd_a1_mean = mean(psd(:,:,1:weeks(2)-1),3);  % average across 4 trials
    psd_a1_SE = std(psd(:,:,1:weeks(2)-1),0,3); % standard deviation across 4 trials
    psd_a1_SE = psd_a1_SE/sqrt(weeks(2)-1); % standard error across 4 trials
    % calculate bounderies for significance testing
    lowlim = psd_a1_mean-2*psd_a1_SE;
    highlim = psd_a1_mean+2*psd_a1_SE;

    % iteration over channels and subs to determine significant changes
    for ch = 1:nchannels
        for b = 1:nsubs
            p=1;
            for i =1:length(weeks)-1
                psd_tempcopy = psd(ch,b,weeks(i):weeks(i+1)-1); % copy data of phase p
                error = substd(ch,b,weeks(i):weeks(i+1)-1);   % copy st.dev. of phase p
                % test negative changes for significance
                psd_temp = psd_tempcopy+error; % add st.dev. to each trial
                psd_temp(psd_temp>=lowlim(ch,b)) = 0; % set unsignificant changes to zero
                psd_temp_sig = -logical(psd_temp); % set significant neg. changes to -1
                % test positive changes for significance
                psd_temp = psd_tempcopy-error;  % subtract std.dev. from each trial
                psd_temp(psd_temp<=highlim(ch,b)) = 0; % set unsignificant changes to zero
                psd_temp = logical(psd_temp); % set significant pos. changes to 1
                psd_temp_sig = psd_temp_sig + psd_temp; % comsube neg. and pos. changes in matrix
                psd_sig_sum(ch,b,p) = sum(psd_temp_sig(:)); % sum the significant scores in phase
                psd_mean(ch,b,p) = mean(psd_tempcopy(:)); % take the mean value of phase
                psd_change(ch,b,p) = (mean(psd_tempcopy(:))-psd_a1_mean(ch,b))...
                    ./psd_a1_mean(ch,b)*100; % calculate percentage change in respect to baseline
                p=p+1;
            end
        end
    end

    % zero s-values outside of interval [-2,2]
    psd_sig_sum(psd_sig_sum>-2 & psd_sig_sum<2) = 0;

    % return result
    psd_sig = psd_sig_sum;

    % return significant changes percentage change score
    psd_sig_mag = logical(psd_sig_sum).*(psd_change);
```

4.3.3 Function: test22SD(...)

```
% Grand Valley State University, Master of Science in Engineering
% ========================================================================
% author: Nadina Zweifel
% date: 3/16/2016
% advisor: Dr. Samhita Rhodes
%
% modified by: Joshua Usoro
% date: 04/03/2017
% advisor: Dr. Samhita Rhodes
% ------------------------------------------------------------------------
% function: [psd_sig,psd_sig_mag,psd_mean,psd_change] = test2SD(psd1,psd2,substd2)
% this function applies the two-standard deviation method to the input
% data which contains data of different phases (baseline and intervention).
% The function assumes that the phases consists of 4 data points and
% test for significance in respect of the first dataset. Significance is tested by
% counting the data points exceeding the 2*SE band (SE=standard error).
% Cases that exceed the band 2 times or more are considered as significant.
% input: psd1, psd2, substd2
% psd1 and psd2 represent the power spectral density of the EEG data under
% consideration. The input data is subned by taking the average across
% data points in a particular frequency range. Hence, psd1 and psd2 contain the
% means of the subs while substd2 contains the standard deviation of each
% sub in the second dataset psd2.
% output: psd_sig, psd_sig_mag, psd_mean, psd_change
% psd_sig is a matrix containing the significance score (s-value) for each
% sub and channel in respect of the baseline. Unsignificant changes are
% represented with a zero. The s-value is 2 or higher which is considered
% as significant. psd_sig_mag is of the same type as psd_sig but instead of
% s-values it contains the percentage change score for signifcant changes.
% psd_mean returns just the mean value for each sub and channel for each
% phase. psd_change contains the percentage change score without
% significance testing.
% -------------------------------------------------------------------------

function [psd_sig,psd_sig_mag,psd2_mean,psd_change] = test22SDtest(psd1,psd2,substd2,weeks)
    [nchannels,nsubs,ntrials] = size(psd1);

     % iteration over channels and subs to determine significant changes
     for chn = 1:nchannels
        for sub = 1:nsubs
            phase=1;
            for i =1:length(weeks)-1

                % calculate the baseline values in phase p of first dataset
                psd1_mean = mean(psd1(chn,sub,weeks(i):weeks(i+1)-1),3); % average across 4
trials
                psd1_SE = std(psd1(chn,sub,weeks(i):weeks(i+1)-1),0,3); % st.dev. across 4 trials
                psd1_SE = psd1_SE/sqrt(weeks(i+1)-weeks(i)); % standard error across 4 trials

                % calculate bounderies for significance testing
                lowlim = psd1_mean-2*psd1_SE;
                highlim = psd1_mean+2*psd1_SE;

                psd2_tempcopy = psd2(chn,sub,weeks(i):weeks(i+1)-1); % copy phase p of second
dataset
                error = substd2(chn,sub,weeks(i):weeks(i+1)-1); % copy error of phase p of second
dataset
                % test negative changes for significance
                psd2_temp = psd2_tempcopy+error; % add st.dev. to each trial
                psd2_temp(psd2_temp>=lowlim) = 0; % set unsignificant changes to zero
                psd2_temp_sig = -logical(psd2_temp); % set significant neg. changes to -1
                % test positive changes for significance
                psd2_temp = psd2_tempcopy-error;  % subtract std.dev. from each trial
                psd2_temp(psd2_temp<=highlim) = 0; % set unsignificant changes to zero
                psd2_temp = logical(psd2_temp); % set significant pos. changes to 1
                psd2_temp_sig = psd2_temp_sig + psd2_temp; % comsube neg. and pos. changes in
matrix
                psd_sig_sum(chn,sub,phase) = sum(psd2_temp_sig(:)); % sum the significant scores
in phase
                psd2_mean(chn,sub,phase) = mean(psd2_tempcopy(:)); % take the mean value of phase
                psd_change(chn,sub,phase) = (mean(psd2_tempcopy(:))-psd1_mean)...
                    ./psd1_mean*100; % calculate percentage change in respect to baseline
                phase=phase+1;
            end
        end
    end

    % zero s-values outside of interval [-2,2]
    psd_sig_sum(psd_sig_sum>-2 & psd_sig_sum<2) = 0;

    % return result
    psd_sig = psd_sig_sum;

    % return significant changes percentage change score
    psd_sig_mag = logical(psd_sig_sum).*(psd_change);
```

4.4 Code: Coherence Analysis

4.4.1 Main Script: MSCanalysis.m

```
% Grand Valley State University, Master of Science in Engineering
% =========================================================================
% author: Joshua Usoro
% date: 04/03/2017
% advisor: Dr. Samhita Rhodes
% -------------------------------------------------------------------------
% Master Thesis: Main script for magnitude-squared coherence analysis of EEG data.
% The script loads the data and calculates the auto psd and cross psd to
% determine coherence between two time series
% -------------------------------------------------------------------------


clear all; %clear workspace
close all; %close figure windows
clc; %clear command window
```

Calculate MSC for each session

```
%load('P7newdata.mat') %load in data
fs = 128; % sampling frequency

% selet 'weeks' and EEG ranges corresponding to the subject
% weeks = [1,5,10,15,20]; %p5-start week for each phase
% theta_range = [2,5]; %p5-theta range
% alpha_range = [6,12]; %p5-alpha range
% weeks = [1,6,11,16,21]; %p6-start week for each phase
% theta_range = [2,5]; %p6-theta range
% alpha_range = [6,11]; %p6-alpha range
weeks = [1,4,9,12,17]; %p7-start week for each phase
theta_range = [2,5]; %p7-theta range
alpha_range = [6,11]; %p7-alpha range

% get number of sessions
ntrials = size(data_rest1,2);

% compute MSC for every trial during each task
for trial = 1:ntrials
    % no interact 1
    [c_theta_MSC_rest1{trial},c_alpha_MSC_rest1{trial}] = singleMSC(data_rest1{trial},fs,...
        theta_range,alpha_range);
    % no interact 2
    [c_theta_MSC_rest2{trial},c_alpha_MSC_rest2{trial}] = singleMSC(data_rest2{trial},fs,...
        theta_range,alpha_range);
    % interaction
    [c_theta_MSC_interact{trial},c_alpha_MSC_interact{trial}] =
singleMSC(data_interact{trial},fs,...
        theta_range,alpha_range);
end

% transform the cell arrays to matrices
theta_MSC_rest1 = (cat(3,c_theta_MSC_rest1{:}));
theta_MSC_rest2 = (cat(3,c_theta_MSC_rest2{:}));
theta_MSC_interact = (cat(3,c_theta_MSC_interact{:}));
alpha_MSC_rest1 = (cat(3,c_alpha_MSC_rest1{:}));
alpha_MSC_rest2 = (cat(3,c_alpha_MSC_rest2{:}));
alpha_MSC_interact = (cat(3,c_alpha_MSC_interact{:}));
```

Calculate significant MSC for No Interact 1

```
[t_msc_sig,t_msc_sig_mag,t_msc_change] = sigScore(theta_MSC_rest1,weeks);
[a_msc_sig,a_msc_sig_mag,a_msc_change] = sigScore(alpha_MSC_rest1,weeks);
```

Display results for No Interact 1

```
% define channel labels
chnlabels = {'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'};

% get number of channels and subs
[nchannels,nsubs] = size(t_msc_sig(:,:,1));

% display results with significant score

% define margin around subplot
margin = 0.06;

% define paramter for colorbar
zmin = -4.5;
```

```
zmax = 4.5;
cb_name = 's-value';
% define resolution colormap: continuous -> ncol=0
ncol = 8;

figure
% display theta change A1-B1
h=subplot(231);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(t_msc_sig(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Theta Significance: A1->B1 (no interaction 1)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display theta change A1-A2
h=subplot(232);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(t_msc_sig(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Theta Significance: A1->A2 (no interaction 1)')
xlabel('Channels')
pbaspect([1 1 1])

% display theta change A1-B2
h=subplot(233);
% adjust margins of subplot
```

```
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(t_msc_sig(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Theta Significance: A1->B2 (no interaction 1)')
xlabel('Channels')
pbaspect([1 1 1])

% display alpha change A1-B1
h=subplot(234);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Alpha Significance: A1->B1 (no interaction 1)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display alpha change A1-A2
h=subplot(235);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Alpha Significance: A1->A2 (no interaction 1)')
xlabel('Channels')
pbaspect([1 1 1])

% display alpha change A1-B2
h=subplot(236);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Alpha Significance: A1->B2 (no interaction 1)')
xlabel('Channels')
pbaspect([1 1 1])

% display results with significant magnitude
% set colorbar properties
zmin = -300;
zmax = 300;
cb_name = 'Change (%)';
% define resolution colormap: continuous -> ncol=0
ncol = 0;

figure
% display theta results for A1-B1
h=subplot(231);
% adjust margin of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(t_msc_sig_mag(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Theta Change MSC: A1->B1 (no interaction 1)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display theta results for A1-A2
h=subplot(232);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% dislay result matrix
imagesc(t_msc_sig_mag(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Theta Change MSC: A1->A2 (no interaction 1)')
xlabel('Channels')
pbaspect([1 1 1])

% display theta results for A1-B2
h=subplot(233);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(t_msc_sig_mag(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Theta Change MSC: A1->B2 (no interaction 1)')
xlabel('Channels')
pbaspect([1 1 1])
```

```
% display theta results for A1-B1
h=subplot(234);
% adjust margin of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig_mag(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Alpha Change MSC: A1->B1 (no interaction 1)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display alpha results for A1-A2
h=subplot(235);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% dislay result matrix
imagesc(a_msc_sig_mag(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Alpha Change MSC: A1->A2 (no interaction 1)')
xlabel('Channels')
pbaspect([1 1 1])

% display alpha results for A1-B2
h=subplot(236);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig_mag(:,:,4))
% set colorbar
```

```
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Alpha Change MSC: A1->B2 (no interaction 1)')
xlabel('Channels')
pbaspect([1 1 1])
```

Calculate significant MSC for No Interact 2

```
[t_msc_sig,t_msc_sig_mag,t_msc_change] = sigScore(theta_MSC_rest2,weeks);
[a_msc_sig,a_msc_sig_mag,a_msc_change] = sigScore(alpha_MSC_rest2,weeks);
```

Display results for No Interact 2

```
% define channel labels
chnlabels = {'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'};

% get number of channels and subs
[nchannels,nsubs] = size(t_msc_sig(:,:,1));

% display results with significant score

% define margin around subplot
margin = 0.06;

% define paramter for colorbar
zmin = -4.5;
zmax = 4.5;
cb_name = 's-value';
% define resolution colormap: continuous -> ncol=0
ncol = 8;

figure
% display theta change A1-B1
h=subplot(231);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(t_msc_sig(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Theta Significance: A1->B1 (no interaction 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display theta change A1-A2
h=subplot(232);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(t_msc_sig(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Theta Significance: A1->A2 (no interaction 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display theta change A1-B2
h=subplot(233);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(t_msc_sig(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Theta Significance: A1->B2 (no interaction 2)')
xlabel('Channels')
pbaspect([1 1 1])
```

```
% display alpha change A1-B1
h=subplot(234);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Alpha Significance: A1->B1 (no interaction 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display alpha change A1-A2
h=subplot(235);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Alpha Significance: A1->A2 (no interaction 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display alpha change A1-B2
h=subplot(236);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig(:,:,4))
% set colorbar
```

```
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Alpha Significance: A1->B2 (no interaction 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display results with significant magnitude
% set colorbar properties
zmin = -300;
zmax = 300;
cb_name = 'Change (%)';
% define resolution colormap: continuous -> ncol=0
ncol = 0;

figure
% display theta results for A1-B1
h=subplot(231);
% adjust margin of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(t_msc_sig_mag(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Theta Change MSC: A1->B1 (no interaction 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display theta results for A1-A2
h=subplot(232);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% dislay result matrix
imagesc(t_msc_sig_mag(:,:,3))
% set colorbar
```

```
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Theta Change MSC: A1->A2 (no interaction 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display theta results for A1-B2
h=subplot(233);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(t_msc_sig_mag(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Theta Change MSC: A1->B2 (no interaction 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display theta results for A1-B1
h=subplot(234);
% adjust margin of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig_mag(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Alpha Change MSC: A1->B1 (no interaction 2)')
```

```
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display alpha results for A1-A2
h=subplot(235);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% dislay result matrix
imagesc(a_msc_sig_mag(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Alpha Change MSC: A1->A2 (no interaction 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display alpha results for A1-B2
h=subplot(236);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig_mag(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Alpha Change MSC: A1->B2 (no interaction 2)')
xlabel('Channels')
pbaspect([1 1 1])
```

Calculate significant MSC for Interaction

```
[t_msc_sig,t_msc_sig_mag,t_msc_change] = sigScore(theta_MSC_interact,weeks);
[a_msc_sig,a_msc_sig_mag,a_msc_change] = sigScore(alpha_MSC_interact,weeks);
```

Display results for Interaction

```
% define channel labels
chnlabels = {'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'};

% get number of channels and subs
[nchannels,nsubs] = size(t_msc_sig(:,:,1));

% display results with significant score

% define margin around subplot
margin = 0.06;

% define paramter for colorbar
zmin = -4.5;
zmax = 4.5;
cb_name = 's-value';
% define resolution colormap: continuous -> ncol=0
ncol = 8;

figure
% display theta change A1-B1
h=subplot(231);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(t_msc_sig(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Theta Significance: A1->B1 (interact)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display theta change A1-A2
h=subplot(232);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(t_msc_sig(:,:,3))
% set colorbar
```

```
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Theta Significance: A1->A2 (interact)')
xlabel('Channels')
pbaspect([1 1 1])

% display theta change A1-B2
h=subplot(233);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(t_msc_sig(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Theta Significance: A1->B2 (interact)')
xlabel('Channels')
pbaspect([1 1 1])

% display alpha change A1-B1
h=subplot(234);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Alpha Significance: A1->B1 (interact)')
```

```
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display alpha change A1-A2
h=subplot(235);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Alpha Significance: A1->A2 (interact)')
xlabel('Channels')
pbaspect([1 1 1])

% display alpha change A1-B2
h=subplot(236);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Alpha Significance: A1->B2 (interact)')
xlabel('Channels')
pbaspect([1 1 1])

% display results with significant magnitude
% set colorbar properties
zmin = -300;
zmax = 300;
cb_name = 'Change (%)';
% define resolution colormap: continuous -> ncol=0
```

```
ncol = 0;

figure
% display theta results for A1-B1
h=subplot(231);
% adjust margin of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(t_msc_sig_mag(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Theta Change MSC: A1->B1 (interact)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display theta results for A1-A2
h=subplot(232);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% dislay result matrix
imagesc(t_msc_sig_mag(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Theta Change MSC: A1->A2 (interact)')
xlabel('Channels')
pbaspect([1 1 1])

% display theta results for A1-B2
h=subplot(233);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
```

```
% display result matrix
imagesc(t_msc_sig_mag(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Theta Change MSC: A1->B2 (interact)')
xlabel('Channels')
pbaspect([1 1 1])

% display theta results for A1-B1
h=subplot(234);
% adjust margin of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig_mag(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Alpha Change MSC: A1->B1 (interact)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display alpha results for A1-A2
h=subplot(235);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% dislay result matrix
imagesc(a_msc_sig_mag(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
```

```
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Alpha Change MSC: A1->A2 (interact)')
xlabel('Channels')
pbaspect([1 1 1])

% display alpha results for A1-B2
h=subplot(236);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig_mag(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Alpha Change MSC: A1->B2 (interact)')
xlabel('Channels')
pbaspect([1 1 1])
```

Calculate Significant MSC for no interact1/2

```
[t_msc_sig,t_msc_sig_mag,t_msc_change] = sigScore2(theta_MSC_rest1,theta_MSC_rest2,weeks);
[a_msc_sig,a_msc_sig_mag,a_msc_change] = sigScore2(alpha_MSC_rest1,alpha_MSC_rest2,weeks);
```

Display results for no interact1/2

```
% set margin parameter
margin = 0.04;

% display results for significance
% set colorbar parameters
zmin = -4.5;
zmax = 4.5;
cb_name = 's';
% define resolution colormap: continuous -> ncol=0
ncol = 8;

figure
% display results for theta A1(no interact 1) - A2(no interact 2)
h=subplot(241);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
```

```
set(h, 'pos', p);
% display result matrix
imagesc(t_msc_sig(:,:,1))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Theta Significance: A1(NI 1)->A1(NI 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for theta B1(no interact 1) - B1(no interact 2)
h=subplot(242);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(t_msc_sig(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Theta Significance: B1(NI 1)->B1(NI 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display results for theta A2(no interact 1) - A2(no interact 2)
h=subplot(243);
% set margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(t_msc_sig(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Theta Significance: A2(NI 1)->A2(NI 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display results for theta B2(no interact 1) - B2(no interact 2)
h=subplot(244);
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
imagesc(t_msc_sig(:,:,4))
% set color bar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Theta Significance: B2(NI 1)->B2(NI 2)')
xlabel('Channels')
pbaspect([1 1 1])

h=subplot(245);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig(:,:,1))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Alpha Significance: A1(NI 1)->A1(NI 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for alpha B1(no interact 1) - B1(no interact 2)
h=subplot(246);
% adjust margins for subplot
p = get(h, 'pos');
```

```
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Alpha Significance: B1(NI 1)->B1(NI 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display results for alpha A2(no interact 1) - A2(no interact 2)
h=subplot(247);
% set margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Alpha Significance: A2(NI 1)->A2(NI 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display results for alpha B2(no interact 1) - B2(no interact 2)
h=subplot(248);
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
imagesc(a_msc_sig(:,:,4))
% set color bar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MSC Alpha Significance: B2(NI 1)->B2(NI 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display significant change in magnitude
% adjust colorbar parameters
zmin = -300;
zmax = 300;
cb_name = 'Change (%)';
% define resolution colormap: continuous -> ncol=0
ncol = 0;

figure
% display results for theta A1(no interact 1) - A1(no interact 2)
h=subplot(241);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(t_msc_sig_mag(:,:,1))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Theta Change MSC: A1(NI 1)->A1(NI 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for theta B1(no interact 1) - B1(no interact 2)
h=subplot(242);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display results matrix
imagesc(t_msc_sig_mag(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Theta Change MSC: B1(NI 1)->B1(NI 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display results for theta A2(no interact 1) - A2(no interact 2)
h=subplot(243);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display results matrix
imagesc(t_msc_sig_mag(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Theta Change MSC: A2(NI 1)->A2(NI 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display results for theta B2(no interact 1) -> B2(no interact 2)
h=subplot(244);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(t_msc_sig_mag(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Theta Change MSC: B2(NI 1)->B2(NI 2)')
xlabel('Channels')
pbaspect([1 1 1])

h=subplot(245);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
```

```
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig_mag(:,:,1))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Alpha Change MSC: A1(NI 1)->A1(NI 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for alpha B1(no interact 1) - B1(no interact 2)
h=subplot(246);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display results matrix
imagesc(a_msc_sig_mag(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Alpha Change MSC: B1(NI 1)->B1(NI 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display results for alpha A2(no interact 1) - A2(no interact 2)
h=subplot(247);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display results matrix
imagesc(a_msc_sig_mag(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Alpha Change MSC: A2(NI 1)->A2(NI 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display results for alpha B2(no interact 1) -> B2(no interact 2)
h=subplot(248);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(a_msc_sig_mag(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Alpha Change MSC: B2(NI 1)->B2(NI 2)')
xlabel('Channels')
pbaspect([1 1 1])
```

4.4.2 Function: singleMSC(…)

```
function [tMSC, aMSC] = singleMSC(input,fs,theta_range,alpha_range)
%function [tMSC, aMSC] = singleMSC(input,fs,theta_range,alpha_range)
%
% This function returns the magnitude-squared coherence between two signals
%
% input: time series for one condition for one trial
% fs: sampling frequency
% theta_range: vector of the start and end of the theta range, e.b. [1,5]
% alpha_range: vector of the start and end of the alpha range, e.b. [6,12]
%
%
%Author: Joshua Usoro
%Date: 04/03/2017

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% go through every electrode combination
for q=1:14
    for r=1:14
```

```
        cxy = mscohere(input(q,:),input(r,:),[],[],[],fs); %calculate MSC
        tMSC(q,r) = mean(cxy(theta_range(1):theta_range(2))); %separate out the theta range
        aMSC(q,r) = mean(cxy(alpha_range(1):alpha_range(2))); %separate out the alpha range
    end
end


end
```

4.5 Code: Mutual Information Analysis

4.5.1 Main Script: MIanalysis.m

```
% Grand Valley State University, Master of Science in Engineering
% ========================================================================
% author: Joshua Usoro
% date: 04/03/2017
% advisor: Dr. Samhita Rhodes
% ------------------------------------------------------------------------
% Master Thesis: Main script for mutual information analysis of EEG data.
% The script loads the data and computes the probability density functions
% to quantify mutual information between two time series
% ------------------------------------------------------------------------

clear all; %clear workspace
close all; %close figures
clc; %clear command window
```

Calculate single MI for each session

```
%load('P7newdata.mat') %load data

% select the 'weeks' that corresponds to the subject
%weeks = [1,5,10,15,20]; %p5-start week for each phase
%weeks = [1,6,11,16,21]; %p6-start week for each phase
weeks = [1,4,9,12,17]; %p7-start week for each phase

% get number of sessions
ntrials = size(data_rest1,2);

% compute mutual information for every trial during each task
for trial = 1:ntrials
    % resting 1
    [c_MI_rest1{trial},c_MImax_rest1{trial}] = singleMI(data_rest1{trial});
    % resting 2
    [c_MI_rest2{trial},c_MImax_rest2{trial}] = singleMI(data_rest2{trial});
    % interaction
    [c_MI_interact{trial},c_MImax_interact{trial}] = singleMI(data_interact{trial});
end
```

```
% transform the cell arrays to matrices
MI_rest1 = (cat(3,c_MI_rest1{:}));
MI_rest2 = (cat(3,c_MI_rest2{:}));
MI_interact = (cat(3,c_MI_interact{:}));
```

Calculate significant MI for No Interact 1

```
[mi_sig,mi_sig_mag,mi_change] = sigScore(MI_rest1,weeks);
```

Display results for No Interact 1

```
% define channel labels
chnlabels = {'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'};

% get number of channels and subs
[nchannels,nsubs] = size(mi_sig(:,:,1));

% display results with significant score

% define margin around subplot
margin = 0.06;

% define paramter for colorbar
zmin = -4.5;
zmax = 4.5;
cb_name = 's-value';
% define resolution colormap: continuous -> ncol=0
ncol = 8;

figure
% display change A1-B1
h=subplot(131);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(mi_sig(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MI Significance: A1->B1 (no interaction 1)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])
```

```
% display change A1-A2
h=subplot(132);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(mi_sig(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MI Significance: A1->A2 (no interaction 1)')
xlabel('Channels')
pbaspect([1 1 1])

% display change A1-B2
h=subplot(133);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(mi_sig(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MI Significance: A1->B2 (no interaction 1)')
xlabel('Channels')
pbaspect([1 1 1])

% display results with significant magnitude
% set colorbar properties
zmin = -300;
zmax = 300;
cb_name = 'Change (%)';
% define resolution colormap: continuous -> ncol=0
ncol = 0;

figure
```

```
% display results for A1-B1
h=subplot(131);
% adjust margin of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(mi_sig_mag(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change MI: A1->B1 (no interaction 1)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for A1-A2
h=subplot(132);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% dislay result matrix
imagesc(mi_sig_mag(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change MI: A1->A2 (no interaction 1)')
xlabel('Channels')
pbaspect([1 1 1])

% display results for A1-B2
h=subplot(133);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(mi_sig_mag(:,:,4))
% set colorbar
```

```
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change MI: A1->B2 (no interaction 1)')
xlabel('Channels')
pbaspect([1 1 1])
```

Calculate significant MI for No Interact 2

```
[mi_sig,mi_sig_mag,mi_change] = sigScore(MI_rest2,weeks);
```

Display results for No Interact 2

```
% define channel labels
chnlabels = {'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'};

% get number of channels and subs
[nchannels,nsubs] = size(mi_sig(:,:,1));

% display results with significant score

% define margin around subplot
margin = 0.06;

% define paramter for colorbar
zmin = -4.5;
zmax = 4.5;
cb_name = 's-value';
% define resolution colormap: continuous -> ncol=0
ncol = 8;

figure
% display change A1-B1
h=subplot(131);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(mi_sig(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MI Significance: A1->B1 (no interaction 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display change A1-A2
h=subplot(132);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(mi_sig(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MI Significance: A1->A2 (no interaction 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display change A1-B2
h=subplot(133);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(mi_sig(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MI Significance: A1->B2 (no interaction 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display results with significant magnitude
% set colorbar properties
zmin = -300;
zmax = 300;
cb_name = 'Change (%)';
% define resolution colormap: continuous -> ncol=0
ncol = 0;

figure
% display results for A1-B1
h=subplot(131);
% adjust margin of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(mi_sig_mag(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change MI: A1->B1 (no interaction 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for A1-A2
h=subplot(132);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% dislay result matrix
imagesc(mi_sig_mag(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change MI: A1->A2 (no interaction 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display results for A1-B2
```

```
h=subplot(133);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(mi_sig_mag(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change MI: A1->B2 (no interaction 2)')
xlabel('Channels')
pbaspect([1 1 1])
```

Calculate significant MI for Interaction

```
[mi_sig,mi_sig_mag,mi_change] = sigScore(MI_interact,weeks);
```

Display results for Interaction

```
% define channel labels
chnlabels = {'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'};

% get number of channels and subs
[nchannels,nsubs] = size(mi_sig(:,:,1));

% display results with significant score

% define margin around subplot
margin = 0.06;

% define paramter for colorbar
zmin = -4.5;
zmax = 4.5;
cb_name = 's-value';
% define resolution colormap: continuous -> ncol=0
ncol = 8;

figure
% display change A1-B1
h=subplot(131);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
```

```
% display result matrix
imagesc(mi_sig(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MI Significance: A1->B1 (interact)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display change A1-A2
h=subplot(132);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(mi_sig(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MI Significance: A1->A2 (interact)')
xlabel('Channels')
pbaspect([1 1 1])

% display change A1-B2
h=subplot(133);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(mi_sig(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
```

```
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MI Significance: A1->B2 (interact)')
xlabel('Channels')
pbaspect([1 1 1])

% display results with significant magnitude
% set colorbar properties
zmin = -300;
zmax = 300;
cb_name = 'Change (%)';
% define resolution colormap: continuous -> ncol=0
ncol = 0;

figure
% display results for A1-B1
h=subplot(131);
% adjust margin of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(mi_sig_mag(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change MI: A1->B1 (interact)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for A1-A2
h=subplot(132);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% dislay result matrix
imagesc(mi_sig_mag(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
```

```
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change MI: A1->A2 (interact)')
xlabel('Channels')
pbaspect([1 1 1])

% display results for A1-B2
h=subplot(133);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(mi_sig_mag(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change MI: A1->B2 (interact)')
xlabel('Channels')
pbaspect([1 1 1])
```

Calculate Significant MI for no interact1/2

```
[mi_sig,mi_sig_mag,mi_change] = sigScore2(MI_rest1,MI_rest2,weeks);
```

Display results for no interact1/2

```
% set margin parameter
margin = 0.04;

% display results for significance
% set colorbar parameters
zmin = -4.5;
zmax = 4.5;
cb_name = 's-value';
% define resolution colormap: continuous -> ncol=0
ncol = 8;

figure
% display results for A1(no interact 1) - A2(no interact 2)
h=subplot(221);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
```

```
% display result matrix
imagesc(mi_sig(:,:,1))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MI Significance: A1(no interact 1)->A1(no interact 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for B1(no interact 1) - B1(no interact 2)
h=subplot(222);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(mi_sig(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MI Significance: B1(no interact 1)->B1(no interact 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for A2(no interact 1) - A2(no interact 2)
h=subplot(223);
% set margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(mi_sig(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MI Significance: A2(no interact 1)->A2(no interact 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for B2(no interact 1) - B2(no interact 2)
h=subplot(224);
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
imagesc(mi_sig(:,:,4))
% set color bar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('MI Significance: B2(no interact 1)->B2(no interact 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display significant change in magnitude
% adjust colorbar parameters
zmin = -300;
zmax = 300;
cb_name = 'Change (%)';
% define resolution colormap: continuous -> ncol=0
ncol = 0;

figure
% display results for A1(no interact 1) - A1(no interact 2)
h=subplot(221);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(mi_sig_mag(:,:,1))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
```

```
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change MI: A1(no interact 1)->A1(no interact 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for B1(no interact 1) - B1(no interact 2)
h=subplot(222);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display results matrix
imagesc(mi_sig_mag(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change MI: B1(no interact 1)->B1(no interact 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for A2(no interact 1) - A2(no interact 2)
h=subplot(223);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display results matrix
imagesc(mi_sig_mag(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change MI: A2(no interact 1)->A2(no interact 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])
```

```
% display results for B2(no interact 1) -> B2(no interact 2)
h=subplot(224);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(mi_sig_mag(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change MI: B2(no interact 1)->B2(no interact 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])
```

4.5.2 Function: singleMI(…)

```
function [normMI, selfMI] = singleMI(data)
%function [normMI, selfMI] = singleMI(data)
%
%This function returns a heat map of the mutual information between EEG
%channels, as well as the matrices of values for the heat map and the
%diagonal (which is represented by zeros in the heat map)
%
% data: single time series for one condition
%Author: Joshua Usoro
%Date: 04/03/2017

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% go through each electrode pair to calculate MI
for j=1:14
    for k=1:14
        bins = floor(sqrt(length(data(1,:))/5)); %bin size
        [counts1, ce1] = hist(data(j,:), bins); %histogram of signal 1
        [counts2, ce2] = hist(data(k,:), bins); %histogram of signal 2
        binW = ce1(2)-ce1(1); %bin width
        p1 = (counts1*binW)/sum(counts1*binW); %probability of signal 1
        p2 = (counts2*binW)/sum(counts2*binW); %probabiltiy of signal 2
        H1 = sum(-(p1(p1>0).*(log2(p1(p1>0))))); %entropy for signal 1
        H2 = sum(-(p2(p2>0).*(log2(p2(p2>0))))); %entropy for signal 2
        maxMI = min(H1, H2); %maximum value for MI
        p12_indep = bsxfun(@times, p1.', p2);
        p12_joint = hist3([data(j,:)', data(k,:)'], ...
            [bins, bins])/length(data(1,:)); %joint pdf
        dI12 = p12_joint.*log(p12_joint./p12_indep); % mutual info at each bin
        I12 = nansum(dI12(:)); % sum of all mutual information
        normMI(j,k) = I12/maxMI*100; %MI percentage
        if j==k
            selfMI(j) = normMI(j,k); %vector for diagonal maximal values
            normMI(j,k)=0; %set diagonal to zero
        end
    end
end


end
```

4.6 Code: Transfer Entropy Analysis

4.4.1 Main Script: TEanalysis.m

```
% Grand Valley State University, Master of Science in Engineering
% ========================================================================
% author: Joshua Usoro
% date: 04/03/2017
% advisor: Dr. Samhita Rhodes
% ------------------------------------------------------------------------
% Master Thesis: Main script for transfer entropy analysis of EEG data.
% The script loads the data and calculates the entropy between time series
% and their time-lagged counterparts
% ------------------------------------------------------------------------


clear all; %clear workspace
close all; %close figure windows
clc; %clear command window
```

Calculate transfer entropy for each session

```
%load('P7newdata.mat') %load data
fs = 128; %sampling frequency
t = 2; %time lag in X from present
w = 2; %time lag in Y from present

% select corresponding 'weeks'
%weeks = [1,5,10,15,20]; %p5-start week for each phase
%weeks = [1,6,11,16,21]; %p6-start week for each phase
weeks = [1,4,9,12,17]; %p7-start week for each phase

% get number of sessions
ntrials = size(data_rest1,2);
```

```
% compute TE for every trial during each task
for trial = 1:ntrials
    % resting 1
    [c_TE_rest1{trial}] = singleTE(data_rest1{trial},t,w);
    % resting 2
    [c_TE_rest2{trial}] = singleTE(data_rest2{trial},t,w);
    % interaction
    [c_TE_interact{trial}] = singleTE(data_interact{trial},t,w);
end

% transform the cell arrays to matrices
TE_rest1 = (cat(3,c_TE_rest1{:}));
TE_rest2 = (cat(3,c_TE_rest2{:}));
TE_interact = (cat(3,c_TE_interact{:}));

% calculate the percentage change from resting 1 to interaction
%psd_subs_change = (psd_subs_rest2 - psd_subs_rest1)./psd_subs_rest1*100;
```

Calculate significant TE for No Interact 1

```
[TE_sig,TE_sig_mag,TE_change] = sigScore(TE_rest1,weeks);
```

Display results for No Interact 1

```
% define channel labels
chnlabels = {'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'};

% get number of channels and subs
[nchannels,nsubs] = size(TE_sig(:,:,1));

% display results with significant score

% define margin around subplot
margin = 0.06;

% define paramter for colorbar
zmin = -4.5;
zmax = 4.5;
cb_name = 's-value';
% define resolution colormap: continuous -> ncol=0
ncol = 8;

figure
% display change A1-B1
h=subplot(131);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(TE_sig(:,:,2))
% set colorbar
```

```
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('TE Significance: A1->B1 (no interaction 1)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display change A1-A2
h=subplot(132);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(TE_sig(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('TE Significance: A1->A2 (no interaction 1)')
xlabel('Channels')
pbaspect([1 1 1])

% display change A1-B2
h=subplot(133);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(TE_sig(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
```

```
title('TE Significance: A1->B2 (no interaction 1)')
xlabel('Channels')
pbaspect([1 1 1])

% display results with significant magnitude
% set colorbar properties
zmin = -300;
zmax = 300;
cb_name = 'Change (%)';
% define resolution colormap: continuous -> ncol=0
ncol = 0;

figure
% display results for A1-B1
h=subplot(131);
% adjust margin of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(TE_sig_mag(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change TE: A1->B1 (no interaction 1)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for A1-A2
h=subplot(132);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% dislay result matrix
imagesc(TE_sig_mag(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
```

```
title('Change TE: A1->A2 (no interaction 1)')
xlabel('Channels')
pbaspect([1 1 1])

% display results for A1-B2
h=subplot(133);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(TE_sig_mag(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change TE: A1->B2 (no interaction 1)')
xlabel('Channels')
pbaspect([1 1 1])

% define colormap for TE arrow plot
map = [0,0,1;
       0,0.333333333333333,1;
       0,0.666666666666667,1;
       0,1,1;
       0.333333333333333,1,0.666666666666667;
       0.666666666666667,1,0.333333333333333;
       1,1,0;
       1,0.666666666666667,0;
       1,0.333333333333333,0];
map2 = copper(2); %extract range of colors from copper map
mymap = [map2;map;map2]; %combine to create new map

% create subplots for TE
TEplot(TE_sig(:,:,2),nchannels); %create TE plot for No Interact 1
gu_ax = gca; %get figure handles of No Interact 1 plot
TEplot(TE_sig(:,:,3),nchannels); %create TE plot for No Interact 2
lu_ax = gca; %get figure handles of No Interact 2 plot
TEplot(TE_sig(:,:,4),nchannels); %create TE plot for No Interact 1/2
mu_ax = gca; %get figure handles of No Interact 1/2 plot

F = figure; % New figure
P1 = subplot(1,3,1); %Plot a subplot.
P1_pos = get(P1,'position'); %get its position.
delete(P1) %Delete the subplot
P2 = subplot(1,3,2); %Plot a subplot.
P2_pos = get(P2,'position'); %get its position.
delete(P2); %Delete the subplot
```

```
P3 = subplot(1,3,3); %Plot a subplot.
P3_pos = get(P3,'position'); %get its position.
delete(P3); %Delete the subplot

P = copyobj(gu_ax,F); %Copy the gu_ax to new fig
set(P,'position',P1_pos) %Set its position to the deleted subplot's
P.Title.String = 'TE Significance: A1->B1 (no interact 1)';
P = copyobj(lu_ax,F); %Copy the gu_ax to new fig
set(P,'position',P2_pos) %Set its position to the deleted subplot's
P.Title.String = 'TE Significance: A1->A2 (no interact 1)';
P = copyobj(mu_ax,F); % Copy the gu_ax to new fig
set(P,'position',P3_pos) %Set its position to the deleted subplot's
P.Title.String = 'TE Significance: A1->B2 (no interact 1)';
colormap(mymap) %define colormap
c = colorbar;
c.Limits = [40,215]; %restrict colorbar range to desired colors
c.Label.String = 's-value'; %label colorbar
c.Ticks = [45:20:215]; %create tickmarks
c.TickLabels = [-4:4]; %create tick labels
```

Calculate significant TE for No Interaction 2

```
[TE_sig,TE_sig_mag,TE_change] = sigScore(TE_rest2,weeks);
```

Display results for No Interaction 2

```
% define channel labels
chnlabels = {'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'};

% get number of channels and subs
[nchannels,nsubs] = size(TE_sig(:,:,1));

% display results with significant score

% define margin around subplot
margin = 0.06;

% define paramter for colorbar
zmin = -4.5;
zmax = 4.5;
cb_name = 's-value';
% define resolution colormap: continuous -> ncol=0
ncol = 8;

figure
% display change A1-B1
h=subplot(131);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
```

```
imagesc(TE_sig(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('TE Significance: A1->B1 (no interaction 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display change A1-A2
h=subplot(132);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(TE_sig(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('TE Significance: A1->A2 (no interaction 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display change A1-B2
h=subplot(133);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(TE_sig(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('TE Significance: A1->B2 (no interaction 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display results with significant magnitude
% set colorbar properties
zmin = -300;
zmax = 300;
cb_name = 'Change (%)';
% define resolution colormap: continuous -> ncol=0
ncol = 0;

figure
% display results for A1-B1
h=subplot(131);
% adjust margin of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(TE_sig_mag(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change TE: A1->B1 (no interaction 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for A1-A2
h=subplot(132);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% dislay result matrix
imagesc(TE_sig_mag(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change TE: A1->A2 (no interaction 2)')
xlabel('Channels')
pbaspect([1 1 1])

% display results for A1-B2
h=subplot(133);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(TE_sig_mag(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change TE: A1->B2 (no interaction 2)')
xlabel('Channels')
pbaspect([1 1 1])

% create subplots for TE
TEplot(TE_sig(:,:,2),nchannels); %create TE plot for No Interact 1
gu_ax = gca; %get figure handles of No Interact 1 plot
TEplot(TE_sig(:,:,3),nchannels); %create TE plot for No Interact 2
lu_ax = gca; %get figure handles of No Interact 2 plot
TEplot(TE_sig(:,:,4),nchannels); %create TE plot for No Interact 1/2
mu_ax = gca; %get figure handles of No Interact 1/2 plot

F = figure; % New figure
P1 = subplot(1,3,1); %Plot a subplot.
P1_pos = get(P1,'position'); %get its position.
delete(P1) %Delete the subplot
P2 = subplot(1,3,2); %Plot a subplot.
P2_pos = get(P2,'position'); %get its position.
delete(P2); %Delete the subplot
P3 = subplot(1,3,3); %Plot a subplot.
P3_pos = get(P3,'position'); %get its position.
delete(P3); %Delete the subplot

P = copyobj(gu_ax,F); %Copy the gu_ax to new fig
set(P,'position',P1_pos) %Set its position to the deleted subplot's
P.Title.String = 'TE Significance: A1->B1 (no interact 1)';
P = copyobj(lu_ax,F); %Copy the gu_ax to new fig
set(P,'position',P2_pos) %Set its position to the deleted subplot's
P.Title.String = 'TE Significance: A1->A2 (no interact 1)';
P = copyobj(mu_ax,F); % Copy the gu_ax to new fig
```

```
set(P,'position',P3_pos) %Set its position to the deleted subplot's
P.Title.String = 'TE Significance: A1->B2 (no interact 1)';
colormap(mymap) %define colormap
c = colorbar;
c.Limits = [40,215]; %restrict colorbar range to desired colors
c.Label.String = 's-value'; %label colorbar
c.Ticks = [45:20:215]; %create tickmarks
c.TickLabels = [-4:4]; %create tick labels
```

Calculate significant TE for Interaction

```
[TE_sig,TE_sig_mag,TE_change] = sigScore(TE_interact,weeks);
```

Display results for Interaction

```
% define channel labels
chnlabels = {'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'};

% get number of channels and subs
[nchannels,nsubs] = size(TE_sig(:,:,1));

% display results with significant score

% define margin around subplot
margin = 0.06;

% define paramter for colorbar
zmin = -4.5;
zmax = 4.5;
cb_name = 's-value';
% define resolution colormap: continuous -> ncol=0
ncol = 8;

figure
% display change A1-B1
h=subplot(131);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(TE_sig(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
```

```
title('TE Significance: A1->B1 (interact)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display change A1-A2
h=subplot(132);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(TE_sig(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('TE Significance: A1->A2 (interact)')
xlabel('Channels')
pbaspect([1 1 1])

% display change A1-B2
h=subplot(133);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(TE_sig(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% adjust graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('TE Significance: A1->B2 (interact)')
xlabel('Channels')
pbaspect([1 1 1])

% display results with significant magnitude
% set colorbar properties
zmin = -300;
zmax = 300;
cb_name = 'Change (%)';
```

```matlab
% define resolution colormap: continuous -> ncol=0
ncol = 0;

figure
% display results for A1-B1
h=subplot(131);
% adjust margin of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(TE_sig_mag(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change TE: A1->B1 (interact)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for A1-A2
h=subplot(132);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% dislay result matrix
imagesc(TE_sig_mag(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change TE: A1->A2 (interact)')
xlabel('Channels')
pbaspect([1 1 1])

% display results for A1-B2
h=subplot(133);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
```

```
set(h, 'pos', p);
% display result matrix
imagesc(TE_sig_mag(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change TE: A1->B2 (interact)')
xlabel('Channels')
pbaspect([1 1 1])

% create subplots for TE
TEplot(TE_sig(:,:,2),nchannels); %create TE plot for No Interact 1
gu_ax = gca; %get figure handles of No Interact 1 plot
TEplot(TE_sig(:,:,3),nchannels); %create TE plot for No Interact 2
lu_ax = gca; %get figure handles of No Interact 2 plot
TEplot(TE_sig(:,:,4),nchannels); %create TE plot for No Interact 1/2
mu_ax = gca; %get figure handles of No Interact 1/2 plot

F = figure; % New figure
P1 = subplot(1,3,1); %Plot a subplot.
P1_pos = get(P1,'position'); %get its position.
delete(P1) %Delete the subplot
P2 = subplot(1,3,2); %Plot a subplot.
P2_pos = get(P2,'position'); %get its position.
delete(P2); %Delete the subplot
P3 = subplot(1,3,3); %Plot a subplot.
P3_pos = get(P3,'position'); %get its position.
delete(P3); %Delete the subplot

P = copyobj(gu_ax,F); %Copy the gu_ax to new fig
set(P,'position',P1_pos) %Set its position to the deleted subplot's
P.Title.String = 'TE Significance: A1->B1 (no interact 1)';
P = copyobj(lu_ax,F); %Copy the gu_ax to new fig
set(P,'position',P2_pos) %Set its position to the deleted subplot's
P.Title.String = 'TE Significance: A1->A2 (no interact 1)';
P = copyobj(mu_ax,F); % Copy the gu_ax to new fig
set(P,'position',P3_pos) %Set its position to the deleted subplot's
P.Title.String = 'TE Significance: A1->B2 (no interact 1)';
colormap(mymap) %define colormap
c = colorbar;
c.Limits = [40,215]; %restrict colorbar range to desired colors
c.Label.String = 's-value'; %label colorbar
c.Ticks = [45:20:215]; %create tickmarks
c.TickLabels = [-4:4]; %create tick labels
```

Calculate Significant TE for no interact1/2

```
[TE_sig,TE_sig_mag,TE_change] = sigScore2(TE_rest1,TE_rest2,weeks);
```

Display results for no interact1/2

```
% set margin parameter
margin = 0.04;

% display results for significance
% set colorbar parameters
zmin = -4.5;
zmax = 4.5;
cb_name = 's-value';
% define resolution colormap: continuous -> ncol=0
ncol = 8;

figure
% display results for A1(no interact 1) - A2(no interact 2)
h=subplot(221);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(TE_sig(:,:,1))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('TE Significance: A1(no interact 1)->A1(no interact 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for B1(no interact 1) - B1(no interact 2)
h=subplot(222);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(TE_sig(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('TE Significance: B1(no interact 1)->B1(no interact 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for A2(no interact 1) - A2(no interact 2)
h=subplot(223);
% set margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(TE_sig(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('TE Significance: A2(no interact 1)->A2(no interact 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for B2(no interact 1) - B2(no interact 2)
h=subplot(224);
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
imagesc(TE_sig(:,:,4))
% set color bar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('TE Significance: B2(no interact 1)->B2(no interact 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])
```

```
% display significant change in magnitude
% adjust colorbar parameters
zmin = -300;
zmax = 300;
cb_name = 'Change (%)';
% define resolution colormap: continuous -> ncol=0
ncol = 0;

figure
% display results for A1(no interact 1) - A1(no interact 2)
h=subplot(221);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(TE_sig_mag(:,:,1))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change TE: A1(no interact 1)->A1(no interact 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for B1(no interact 1) - B1(no interact 2)
h=subplot(222);
% adjust margins for subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display results matrix
imagesc(TE_sig_mag(:,:,2))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change TE: B1(no interact 1)->B1(no interact 2)')
xlabel('Channels')
ylabel('Channels')
```

```
pbaspect([1 1 1])

% display results for A2(no interact 1) - A2(no interact 2)
h=subplot(223);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display results matrix
imagesc(TE_sig_mag(:,:,3))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change TE: A2(no interact 1)->A2(no interact 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])

% display results for B2(no interact 1) -> B2(no interact 2)
h=subplot(224);
% adjust margins of subplot
p = get(h, 'pos');
p([3 4]) = p([3 4]) + margin;
set(h, 'pos', p);
% display result matrix
imagesc(TE_sig_mag(:,:,4))
% set colorbar
setColorbar(zmin,zmax,cb_name,1,ncol);
% set graph properties
set(gca, 'YDir', 'normal');
set(gca,'XTick',1:nsubs)
set(gca,'XTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
set(gca,'YTick',1:nchannels)
set(gca,'YTickLabel',{'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz' ...
            'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'},'fontsize',8)
title('Change TE: B2(no interact 1)->B2(no interact 2)')
xlabel('Channels')
ylabel('Channels')
pbaspect([1 1 1])
%}

%using subplots for TE
TEplot(TE_sig(:,:,1),nchannels);
gu_ax = gca;
TEplot(TE_sig(:,:,2),nchannels);
lu_ax = gca;
```

```
TEplot(TE_sig(:,:,3),nchannels);
mu_ax = gca;
TEplot(TE_sig(:,:,4),nchannels);
nu_ax = gca;

F = figure; % New figure
P1 = subplot(2,2,1); % Plot a subplot.
P1_pos = get(P1,'position'); % get its position.
delete(P1) % Delete the subplot
P2 = subplot(2,2,2);
P2_pos = get(P2,'position');
delete(P2);
P3 = subplot(2,2,3);
P3_pos = get(P3,'position');
delete(P3);
P4 = subplot(2,2,4);
P4_pos = get(P4,'position');
delete(P4);
P = copyobj(gu_ax,F); % Copy the gu_ax to new fig
set(P,'position',P1_pos) % Set its position to the deleted subplot's
P.Title.String = 'TE Significance: A1(no interact 1)->A1(no interact 2)';
c.Ticks = [45:20:215];
c.TickLabels = [-4:4];
P = copyobj(lu_ax,F);
set(P,'position',P2_pos)
P.Title.String = 'TE Significance: B1(no interact 1)->B1(no interact 2)';
c.Ticks = [45:20:215];
c.TickLabels = [-4:4];
P = copyobj(mu_ax,F); % Copy the gu_ax to new fig
set(P,'position',P3_pos) % Set its position to the deleted subplot's
P.Title.String = 'TE Significance: A2(no interact 1)->A2(no interact 2)';
c.Ticks = [45:20:215];
c.TickLabels = [-4:4];
P = copyobj(nu_ax,F); % Copy the gu_ax to new fig
set(P,'position',P4_pos) % Set its position to the deleted subplot's
P.Title.String = 'TE Significance: B2(no interact 1)->B2(no interact 2)';
colormap(mymap)
c = colorbar;
c.Limits = [40,215];
c.Label.String = 's-value';
c.Ticks = [45:20:215];
c.TickLabels = [-4:4];
c.Position = [0.93 0.175 0.02 0.7];
```

4.6.2 Function: singleTE(…)

```
function [TE] = singleTE(data,t,w)
%function [TE] = singleTE(data,t,w)
%
%This function returns a heat map of the transfer entropy between EEG
%channels
```

```
%
% t: time lag in X from present
% w: time lag in Y from present
% T: transfer entropy (bits)
%
%Author: Joshua Usoro
%Date: 04/03/2017

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% go through electrode pair combinations
for j=1:14
    for k=1:14
        TE(j,k) = transferEntropyPartition(data(j,:),data(k,:),t,w); %calculate TE
    end
end

end
```

4.6.3 Function: TEplot(…)

```
function TEplot(data,nchannels)
%function TEplot(data,nchannels)
%
%This function returns a directional mapping of the transfer entropy between EEG
%channels
%
% data: time series for one condition for one trial
% nchannels: number of channels
%
%Author: Joshua Usoro
%Date: 04/03/2017

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

scalp = rgb2gray(imread('scalp.png')); %read in electrode placement image
scalp = scalp(:,25:end); %resize image
G = graph(); %create empty graph handle
G = addnode(G,14); %create network of nodes

% locations of the electrodes
coord = [75,115,160,205,245,110,160,210,75,115,160,205,245,160;...
        140,145,150,145,140,200,200,200,260,255,250,255,260,300]';

% create custom colormap
map = [0,0,1;
       0,0.333333333333333,1;
       0,0.666666666666667,1;
       0,1,1;
       0.333333333333333,1,0.666666666666667;
       0.666666666666667,1,0.333333333333333;
       1,1,0;
       1,0.666666666666667,0;
       1,0.333333333333333,0];

map1 = [0,0,1;
       0,0,1;
       0,0.333333333333333,1;
       0,0.666666666666667,1;
       0,1,1;
       0.333333333333333,1,0.666666666666667;
       0.666666666666667,1,0.333333333333333;
       1,1,0;
       1,0.666666666666667,0;
       1,0.333333333333333,0
       1,0.333333333333333,0];

map2 = copper(2);
mymap = [map2;map;map2];

% create figure
figure;
imagesc(scalp)
colormap(mymap)
axis off
hold on
h = plot(G); %plot nodes on top of scalp image

% locations for node labels
h.XData = [75,115,160,205,245,110,160,210,75,115,160,205,245,160];
h.YData = [140,145,150,145,140,200,200,200,260,255,250,255,260,300];
h.NodeLabel = {'F7' 'F3' 'Fz' 'F4' 'F8' 'C3' 'Cz'...
    'C4' 'P7' 'P3' 'Pz' 'P4' 'P8' 'Oz'};
h.MarkerSize = 8;
h.NodeColor = 'black';

% go through significant electrode pairs
for y=1:nchannels
    for x=1:nchannels
        if data(x,y)~=0 %if electrode has significance
            if abs(data(x,y)) == 5 %special case in one data set
                delta = coord(x,:)-coord(y,:); %create displacement vector
                %create quiver plot based node locations and displacement
                q = quiver(coord(y,1),coord(y,2),delta(1),delta(2),0);
                q.Color = map1(data(x,y)+6,:); %use color from colormap
                q.MaxHeadSize = 0.4; %resize arrow head
                q.LineWidth = 1.5; %resize error width
            else %performs same function for normal data
                delta = coord(x,:)-coord(y,:);
                q = quiver(coord(y,1),coord(y,2),delta(1),delta(2),0);
                q.Color = map(data(x,y)+5,:);
                q.MaxHeadSize = 0.4;
                q.LineWidth = 1.5;
            end
        end
    end
end

hold off
c = colorbar;
c.Limits = [40,215];
c.Label.String = 's-value';
c.Label.HorizontalAlignment = 'center';
c.Label.VerticalAlignment = 'top';
c.Ticks = [45:20:215];
c.TickLabels = [-4:4];
set(gcf,'Visible','off') %turn off output when calling function


end
```

4.6.4 Function: transferEntropyPartition(…)

```
function [T nPar dimPar]=transferEntropyPartition(X,Y,t,w)

%%%%%
% This function computes the transfer entropy between time series X and Y,
% with the flow of information directed from X to Y. Probability density
% estimation is based on the Darbellay-Vajda partitioning algorithm.
%
% For details, please see T Schreiber, "Measuring information transfer", Physical Review Letters,
85(2):461-464, 2000.
%
% Inputs:
% X: source time series in 1-D vector
% Y: target time series in 1-D vector
% t: time lag in X from present
% w: time lag in Y from present
%
% Outputs:
% T: transfer entropy (bits)
% nPar: number of partitions
% dimPar: 1-D vector containing the length of each partition (same along all three dimensions)
%
%
% Copyright 2011 Joon Lee
%
% This program is free software: you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation, either version 3 of the License, or
% (at your option) any later version.
%
% This program is distributed in the hope that it will be useful,
% but WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
% GNU General Public License for more details.
```

```
%
% You should have received a copy of the GNU General Public License
% along with this program.  If not, see <http://www.gnu.org/licenses/>.
%%%%%


% fix block lengths at 1
l=1; k=1;

X=X(:)';
Y=Y(:)';

% go through the time series X and Y, and populate Xpat, Ypat, and Yt
Xpat=[]; Ypat=[]; Yt=[];
for i=max([l+t k+w]):1:min([length(X) length(Y)])
    Xpat=[Xpat; X(i-l-t+1:i-t)];
    Ypat=[Ypat; Y(i-k-w+1:i-w)];
    Yt=[Yt; Y(i)];
end

% ordinal sampling (ranking)
Nt=length(Xpat);
[B,IX]=sort(Xpat);
Xpat(IX)=1:Nt;
[B,IX]=sort(Ypat);
Ypat(IX)=1:Nt;
[B,IX]=sort(Yt);
Yt(IX)=1:Nt;

% compute transfer entropy
partitions=DVpartition3D(Xpat,Ypat,Yt,1,Nt,1,Nt,1,Nt);
nPar=length(partitions);
dimPar=zeros(nPar,1);
for i=1:nPar
    dimPar(i)=partitions(i).Xmax-partitions(i).Xmin+1;
end
T=0;
for i=1:length(partitions)
    a=partitions(i).N/Nt;
    b=sum(Xpat>=partitions(i).Xmin & Xpat<=partitions(i).Xmax & Ypat>=partitions(i).Ymin &
Ypat<=partitions(i).Ymax)/Nt;
    c=sum(Yt>=partitions(i).Zmin & Yt<=partitions(i).Zmax & Ypat>=partitions(i).Ymin &
Ypat<=partitions(i).Ymax)/Nt;
    d=(partitions(i).Ymax-partitions(i).Ymin+1)/Nt;
    T=T+a*log2((a*d)/(b*c));
end
```

4.6.5 Function: DVpartition3D(…)

```
function partitions=DVpartition3D(X,Y,Z,Xmin,Xmax,Ymin,Ymax,Zmin,Zmax)
```

```
%%%%%
% This function implements the Darbellay-Vajda partitioning algorithm for a
% 3D space in a recursive manner. In order to support recursion, Each
% function call should specify a sub-space where partitioning should be
% executed. It is assumed that X, Y, and Z are ordinal (ranked, starting from 1) samples.
%
% For details, please see G A Darbellay and I Vajda, "Estimation of the information by an
adaptive partitioning of the observation space", IEEE Transactions on Information Theory,
45(4):1315-1321, 1999.
%
% Inputs:
% X: 1-D vector containing coordinates in the first dimension
% Y: 1-D vector containing coordinates in the second dimension
% Z: 1-D vector containing coordinates in the third dimension
% Xmin: lower limit in the first dimension
% Ymin: lower limit in the second dimension
% Zmin: lower limit in the third dimension
% Xmax: upper limit in the first dimension
% Ymax: upper limit in the second dimension
% Zmax: upper limit in the third dimension
%
% Outputs:
% partitions: 1-D structure array that contains the lower and upper limits of each partition in
each dimension as well as the number of data points (N) in the partition
%
%
% Copyright 2011 Joon Lee
%
% This program is free software: you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation, either version 3 of the License, or
% (at your option) any later version.
%
% This program is distributed in the hope that it will be useful,
% but WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
% GNU General Public License for more details.
%
% You should have received a copy of the GNU General Public License
% along with this program.  If not, see <http://www.gnu.org/licenses/>.
%%%%%


alpha=0.05;

idx= X>=Xmin & X<=Xmax & Y>=Ymin & Y<=Ymax & Z>=Zmin & Z<=Zmax;
Xsub=X(idx);
Ysub=Y(idx);
Zsub=Z(idx);

Xdiv=floor(mean([Xmin Xmax]));
Ydiv=floor(mean([Ymin Ymax]));
Zdiv=floor(mean([Zmin Zmax]));
```

```
N=[sum(Xsub<=Xdiv & Ysub<=Ydiv & Zsub<=Zdiv) sum(Xsub>Xdiv & Ysub<=Ydiv & Zsub<=Zdiv)
sum(Xsub<=Xdiv & Ysub>Ydiv & Zsub<=Zdiv) sum(Xsub>Xdiv & Ysub>Ydiv & Zsub<=Zdiv) ...
    sum(Xsub<=Xdiv & Ysub<=Ydiv & Zsub>Zdiv) sum(Xsub>Xdiv & Ysub<=Ydiv & Zsub>Zdiv)
sum(Xsub<=Xdiv & Ysub>Ydiv & Zsub>Zdiv) sum(Xsub>Xdiv & Ysub>Ydiv & Zsub>Zdiv)];
T=sum((mean(N)-N).^2)./mean(N);  % has been corrected to include a normalization factor in the
denominator
partitions=struct('Xmin',{},'Xmax',{},'Ymin',{},'Ymax',{},'Zmin',{},'Zmax',{},'N',{});

if T>icdf('chi2',1-alpha,7) && Xmax~=Xmin && Ymax~=Ymin && Zmax~=Zmin
    if N(1)~=0
        partitions=[partitions DVpartition3D(X,Y,Z,Xmin,Xdiv,Ymin,Ydiv,Zmin,Zdiv)];
    end
    if N(3)~=0
        partitions=[partitions DVpartition3D(X,Y,Z,Xmin,Xdiv,Ydiv+1,Ymax,Zmin,Zdiv)];
    end
    if N(2)~=0
        partitions=[partitions DVpartition3D(X,Y,Z,Xdiv+1,Xmax,Ymin,Ydiv,Zmin,Zdiv)];
    end
    if N(4)~=0
        partitions=[partitions DVpartition3D(X,Y,Z,Xdiv+1,Xmax,Ydiv+1,Ymax,Zmin,Zdiv)];
    end
    if N(5)~=0
        partitions=[partitions DVpartition3D(X,Y,Z,Xmin,Xdiv,Ymin,Ydiv,Zdiv+1,Zmax)];
    end
    if N(7)~=0
        partitions=[partitions DVpartition3D(X,Y,Z,Xmin,Xdiv,Ydiv+1,Ymax,Zdiv+1,Zmax)];
    end
    if N(6)~=0
        partitions=[partitions DVpartition3D(X,Y,Z,Xdiv+1,Xmax,Ymin,Ydiv,Zdiv+1,Zmax)];
    end
    if N(8)~=0
        partitions=[partitions DVpartition3D(X,Y,Z,Xdiv+1,Xmax,Ydiv+1,Ymax,Zdiv+1,Zmax)];
    end
elseif sum(idx)~=0
    partitions(1).Xmin=Xmin;
    partitions(1).Xmax=Xmax;
    partitions(1).Ymin=Ymin;
    partitions(1).Ymax=Ymax;
    partitions(1).Zmin=Zmin;
    partitions(1).Zmax=Zmax;
    partitions(1).N=sum(idx);
end
```

4.7 Main Script: readAndSaveData.m

```
% Grand Valley State University, Master of Science in Engineering
% ======================================================================
% author: Nadina Zweifel
% date: 3/16/2016
% advisor: Dr. Samhita Rhodes
% ----------------------------------------------------------------------
% Master Thesis: Main script for reading the data and randomly pick an
% epoch of at least 4 seconds. The data is saved to a mat-file which can be loaded
% for further analysis.
% -------------------------------------------------------------------------

clear all, close all, clc

% define sample length for randomly picked epoch
nsamples = 17*128;

% define channels to analyze
nchannels = 1:14;

% create empty data matrix
%vars_5.mat = zeros(length(nchannels),nsamples);

% define filenames for resting 1 data
%{
filenames = {'b1_1_tsk1_wIM125_data.mat','b1_2_tsk1_wIM125_data.mat',...
    'b1_3_tsk1_wIM125_data.mat','b1_4_tsk1_wIM125_data.mat',...
    'i1_1_tsk1_wIM125_data.mat','i1_2_tsk1_wIM125_data.mat',...
    'i1_3_tsk1_wIM125_data.mat','i1_4_tsk1_wIM125_data.mat',...
    'b2_1_tsk1_wIM125_data.mat','b2_2_tsk1_wIM125_data.mat',...
    'b2_3_tsk1_wIM125_data.mat','b2_4_tsk1_wIM125_data.mat',...
    'i2_1_tsk1_wIM125_data.mat','i2_2_tsk1_wIM125_data.mat',...
    'i2_3_tsk1_wIM125_data.mat','i2_4_tsk1_wIM125_data.mat'};
%}
filenames = {...
    'P5_B1_W3_T1_data.mat', 'P5_B1_W4_T1_data.mat',...
    'P5_B1_W5_T1_data.mat', 'P5_I1_W1_T1_data.mat',...
    'P5_I1_W2_T1_data.mat', 'P5_I1_W3_T1_data.mat',...
    'P5_I1_W4_T1_data.mat', 'P5_I1_W5_T1_data.mat',...
    'P5_B2_W1_T1_data.mat', 'P5_B2_W2_T1_data.mat',...
    'P5_B2_W3_T1_data.mat',...
    'P5_I2_W1_T1_data.mat', ...
    'P5_I2_W2_T1_data.mat', 'P5_I2_W3_T1_data.mat',...
    'P5_I2_W4_T1_data.mat', 'P5_I2_W5_T1_data.mat',};

% read data files
data_rest1 = readFiles(filenames,nchannels,nsamples);

% define filenames for resting 2 data
%{
filenames = {'b1_1_tsk1b_wIM125_data.mat','b1_2_tsk1b_wIM125_data.mat',...
    'b1_3_tsk1b_wIM125_data.mat','b1_4_tsk1b_wIM125_data.mat',...
    'i1_1_tsk1b_wIM125_data.mat','i1_2_tsk1b_wIM125_data.mat',...
    'i1_3_tsk1b_wIM125_data.mat','i1_4_tsk1b_wIM125_data.mat',...
    'b2_1_tsk1b_wIM125_data.mat','b2_2_tsk1b_wIM125_data.mat',...
    'b2_3_tsk1b_wIM125_data.mat','b2_4_tsk1b_wIM125_data.mat',...
    'i2_1_tsk1b_wIM125_data.mat','i2_2_tsk1b_wIM125_data.mat',...
    'i2_3_tsk1b_wIM125_data.mat','i2_4_tsk1b_wIM125_data.mat'};
%}

nsamples = 16*128;
```

```
filenames = {...
    'P5_B1_W3_T1b_data.mat', 'P5_B1_W4_T1b_data.mat',...
    'P5_B1_W5_T1b_data.mat', 'P5_I1_W1_T1b_data.mat',...
    'P5_I1_W2_T1b_data.mat', 'P5_I1_W3_T1b_data.mat',...
    'P5_I1_W4_T1b_data.mat', 'P5_I1_W5_T1b_data.mat',...
    'P5_B2_W1_T1b_data.mat', 'P5_B2_W2_T1b_data.mat',...
    'P5_B2_W3_T1b_data.mat',...
    'P5_I2_W1_T1b_data.mat', ...
    'P5_I2_W2_T1b_data.mat', 'P5_I2_W3_T1b_data.mat',...
    'P5_I2_W4_T1b_data.mat', 'P5_I2_W5_T1b_data.mat',};

% read data files
data_rest2 = readFiles(filenames,nchannels,nsamples);

% define filenames for interaction data
%{
filenames = {'b1_1_tsk2_wIM125_data.mat','b1_2_tsk2_wIM125_data.mat',...
    'b1_3_tsk2_wIM125_data.mat','b1_4_tsk2_wIM125_data.mat',...
    'i1_1_tsk2_wIM125_data.mat','i1_2_tsk2_wIM125_data.mat',...
    'i1_3_tsk2_wIM125_data.mat','i1_4_tsk2_wIM125_data.mat',...
    'b2_1_tsk2_wIM125_data.mat','b2_2_tsk2_wIM125_data.mat',...
    'b2_3_tsk2_wIM125_data.mat','b2_4_tsk2_wIM125_data.mat',...
    'i2_1_tsk2_wIM125_data.mat','i2_2_tsk2_wIM125_data.mat',...
    'i2_3_tsk2_wIM125_data.mat','i2_4_tsk2_wIM125_data.mat'};
%}

nsamples = 4*128;

filenames = {...
    'P5_B1_W3_T2_data.mat', 'P5_B1_W4_T2_data.mat',...
    'P5_B1_W5_T2_data.mat', 'P5_I1_W1_T2_data.mat',...
    'P5_I1_W2_T2_data.mat', 'P5_I1_W3_T2_data.mat',...
    'P5_I1_W4_T2_data.mat', 'P5_I1_W5_T2_data.mat',...
    'P5_B2_W1_T2_data.mat', 'P5_B2_W2_T2_data.mat',...
    'P5_B2_W3_T2_data.mat',...
    'P5_I2_W1_T2_data.mat', ...
    'P5_I2_W2_T2_data.mat', 'P5_I2_W3_T2_data.mat',...
    'P5_I2_W4_T2_data.mat', 'P5_I2_W5_T2_data.mat',};
% read data files
data_interact = readFiles(filenames,nchannels,nsamples);

% load workspace from a file
%clear all
%load('vars_6.mat')

% save variables to mat-file
%save('P7newdata.mat');
```

4.8 Function: readFiles(…)

```
% Grand Valley State University, Master of Science in Engineering
% =========================================================================
% author: Nadina Zweifel
% date: 3/16/2016
% advisor: Dr. Samhita Rhodes
% -------------------------------------------------------------------------
% function: data = readFiles(filename,nchannels,nsamples)
% input: filename, nchannels, nsamples
% filename is a cell array with all the filenames to be loaded
% nchannels fines the number of channels, while nsamples defines the length
% of the epoch which is randomly picked from each file.
% output: data
% data returns a cell array with the loaded data and represents a randomly
% picked epoch of the available data in the file.
% -------------------------------------------------------------------------

function data = readFiles(filename,nchannels,nsamples)

% get number of files
nfiles = size(filename,2);

% read files and pick random epoch of length nsamples
for nf = 1:nfiles
    file = filename{nf}; % get filename
    load(file); % load file
    [pathstr,name,ext] = fileparts(file); % get variable name
    dataset = eval(name); % get values from variable
    % pick epoch randomly from dataset
    data{nf} = pickRandomEpoch(dataset,nchannels,nsamples); % save data in variable
end

% function to randomly pick an epoch of the input data
function epoch = pickRandomEpoch(data,nchannels,nsamples)

% determine length of input data
Namples = size(data,2);

% define right limit
smpmax = Namples - nsamples;

% prevent exceeding indexes
if smpmax <= 0
    tmin = 1;
else
    % get random index for left limit
    tmin = randi(smpmax,1);
end

% return epoch between left and right limit
epoch = data(nchannels,tmin:tmin+nsamples-1);
```

4.9 Function: psdbinswitherr(…)

```
% Grand Valley State University, Master of Science in Engineering
% ========================================================================
% author: Nadina Zweifel
% date: 3/16/2016
% advisor: Dr. Samhita Rhodes
% ------------------------------------------------------------------------
% function: [pxx,pxx_subs,pxx_error,freq_subs] = psdsubswitherr(data,llimit,hlimit,wsub,norm)
% input: data, llimit, hlimit, wsub, norm
% data is a matrix with the time series where the rows are the channels and
% the columns the samples. llimit and hlimit define the frequency range in
% Hz to be returned from the spectrum. wsub defines the sub with in samples
% while norm is a flag value that determines the normalization method.
% output: pxx, pxx_subs, pxx_error freq_subs
% pxx returns the unsubned spectrum normalized by the method defined in
% norm. pxx_subs returns the spectrum which is subned over the width of wsub
% samples. pxx_error returns the standard error resulting from the sub width
% wsub. freq_subs is frequency vector containing the left frequency value
% in Hz.
% ------------------------------------------------------------------------

function [pxx,freq,pxx_subs,pxx_error,freq_subs] = ...
    psdsubswitherr(data,llimit,hlimit,wsub,norm)

% calculate optimal FFT length
Fs = 128;   % sampling frequency
wlength= 320;   % window length for Welch's method in samples
NFFT = 2^nextpow2(wlength+1); % calculate FFT length

% convert frequency limits to samples
lfsmp = floor(llimit*4);
hfsmp = ceil(hlimit*4);

% determine number of channels
numch = size(data,1);

% compute spectrum iterating over channels using the Welch's method with a
% hanning window of wlength
for ch = 1:numch
    [pxxch,freq] = pwelch(data(ch,:),hanning(wlength),[],NFFT,Fs);
     pxx(ch,:) = pxxch;
end

% normalize by channel mean power if norm=0
if norm==0
    pxx = pxx./repmat(mean(pxx,2),[1 size(pxx,2)]);
%     disp('PSD normalized to channel power before subning');
end

% normalize by overall mean power if norm=1
if norm==1
    pxx = pxx./mean(pxx(:));
```

```
%     disp('PSD normalized to overall power before subning');
end

% compute the mean of each sub of width wsub between the frequency limits
% lfsmp and hfsmp
for ch = 1:numch
    b=1;
    for f = lfsmp:wsub:hfsmp-wsub
        pxx_subs(ch,b) = mean(pxx(ch,f:f+3));
        pxx_error(ch,b) = std(pxx(ch,f:f+3))/sqrt(wsub);
        b=b+1;
    end
end

% normalize to the channel mean power after subning if norm=2
if norm==2
    pxx_subs = pxx_subs./repmat(mean(pxx_subs,2),[1 size(pxx_subs,2)]);
%     disp('PSD normalized to channel power after subning');
end

% normalize to the overall mean power after subning if norm=3
if norm==3
    pxx_subs = pxx_subs/mean(pxx_subs(:));
%     disp('PSD normalized to overall power after subning');
end

% return frequency vector for computed subs
freq_subs = (lfsmp:wsub:hfsmp-wsub)/4;
```

4.10 Function: setColorbar(…)

```
% Grand Valley State University, Master of Science in Engineering
% =======================================================================
% author: Nadina Zweifel
% date: 3/16/2016
% advisor: Dr. Samhita Rhodes
% -----------------------------------------------------------------------
% function: setColorbar(zmin,zmax,cbname,split)
% input: zmin, zmax, cbname, split, ncol
% the inputs zmin and zmax represent the lower and upper limit of the
% colorbar, while cbname is a string used for labeling the colorbar. The
% variable split is either 1 or 0 and is defines whether the zero point of
% the colorbar is set to white (1) or not (0).
% ncol defines the resolution of the colormap with ncol colors, while ncol
% = 0 is high resultion of 8192 colors using the OpenGL renderer.
% -----------------------------------------------------------------------

function setColorbar(zmin,zmax,cbname,split,ncol)

% reset colormap
colormap default
```

```
% create new colormap for change
if ncol
    newmap = jet(ncol+1);
else
    newmap = jet(8191);
    set(gcf,'renderer','OpenGL') % this allows higher resolution for the colors
end

% set colormap limits
caxis([zmin zmax]);
% determine position of the zero point
zeropos = (1-zmax/(zmax-zmin));

% if flag=1 set zero point to white
if split
    ncol = size(newmap,1);
    zpos = ceil(ncol*zeropos); % find zeropoint of colorbar
    newmap(zpos,:) = ones(1,3); %set that position to white
    % activate new colormap
    colormap(newmap);
end

% adjust settings of new colormap
h = colorbar;
set(get(h,'title'),'string',cbname,'fontsize',8);

end
```

4.11 Function: sigScore(...)

```
function [mi_sig,mi_sig_mag,mi_change] = sigScore(mi,weeks)

    [nchannels,nsubs,nsessions] = size(mi);

    % calculate the baseline values in the first phase of the input data
    mi_a1_mean = mean(mi(:,:,1:weeks(2)-1),3);  % average across 4 trials
    mi_a1_SE = std(mi(:,:,1:weeks(2)-1),0,3); % standard deviation across 4 trials
    mi_a1_SE = mi_a1_SE/sqrt(weeks(2)-1); % standard error across 4 trials

    % calculate bounderies for significance testing
    lowlim = mi_a1_mean-2*mi_a1_SE;
    highlim = mi_a1_mean+2*mi_a1_SE;

% iteration over channels and subs to determine significant changes
for ch = 1:nchannels
    for b = 1:nsubs
        p=1;
        for i =1:length(weeks)-1
            mi_tempcopy = mi(ch,b,weeks(i):weeks(i+1)-1); % copy data of phase p
            error = mi_a1_SE(ch,b); % copy std error
            % test negative changes for significance
            mi_temp = mi_tempcopy+error; % add st.dev. to each trial
            mi_temp(mi_temp>=lowlim(ch,b)) = 0; % set unsignificant changes to zero
            mi_temp_sig = -logical(mi_temp); % set significant neg. changes to -1
            % test positive changes for significance
            mi_temp = mi_tempcopy-error;  % subtract std.dev. from each trial
            mi_temp(mi_temp<=highlim(ch,b)) = 0; % set unsignificant changes to zero
            mi_temp = logical(mi_temp); % set significant pos. changes to 1
            mi_temp_sig = mi_temp_sig + mi_temp; % combine neg. and pos. changes in matrix
            mi_sig_sum(ch,b,p) = sum(mi_temp_sig(:)); % sum the significant scores in phase
            mi_mean(ch,b,p) = mean(mi_tempcopy(:)); % take the mean value of phase
            mi_change(ch,b,p) = (mean(mi_tempcopy(:))-mi_a1_mean(ch,b))...
                ./mi_a1_mean(ch,b)*100; % calculate percentage change in respect to baseline
            p=p+1;
        end
    end
end

% zero s-values outside of interval [-2,2]
mi_sig_sum(mi_sig_sum>-2 & mi_sig_sum<2) = 0;

% return result
mi_sig = mi_sig_sum;

% return significant changes percentage change score
mi_sig_mag = logical(mi_sig_sum).*(mi_change);
```

5 Bibliography

1. Zweifel, N. et al, “Changes in EEG Spectra of a Child with Severe Disabilities in Response to Power Mobility Training.”
2. Kenyon, L. et al, “Power Mobility Training for Young Children with Multiple, Severe Impairments: A Case Series.”, 2016, *Phys Occup Ther Pediatr*., 2638 (January): 1-16. DOI:10.3109/01942638.2015.1108380.
3. Kenyon L. et al, “Promoting Self-exploration and Function Through an Individualized Power Mobility Training Program.”, 2015, *Pediatr Phys Ther*., Vol 27(2), 200-206. doi:10.1097/PEP.0000000000000129.
4. Klingberg, T. et al, “Increased Brain Activity in Frontal and Parietal Cortex Underlies the Development of Visuospatial Working Memory Capacity during Childhood*”,* 2002, *Journal of Cognitive Neuroscience*, Vol. 14(1), 1-10. doi:10.1162/089892902317205276
5. Butler, C., “Effects of Powered Mobility on Self-Initiated Behavior of Young Children with Locomotor Disability”, 1986, *Developmental Medicine and Child Neurology*, Vol 28, 325-332
6. Deitz, J. et al, “Powered Mobility and Preschoolers with Complex Developmental Delays”, 2002, *American Journal of Occupational Therapy*, Vol 56, 86-96. DOI:10.5014/ajot.56.1.86
7. Casey, B. et al, “Changes in cerebral functional organization during cognitive development”, 2005, *Current Opinion in Neurobiology*, Vol 15, 239-244
8. Casey, B.J. et al, “Structural and functional brain development and its relation to cognitive development”, 2000, *Biological Psychology*, Vol 54, 241-257

9. Blakemore, C. et al. *The Oxford Companion to The Body* (115-118). New York: Oxford University Press, 2001. Print.
10. Carter, R. et al, *The Human Brain Book* (50-74). New York: DK Publishing, 2014. Print.
11. Lovett, E.G. et al, “Time-frequency coherence analysis of atrial fibrillation termination during procainamide administration”, 1997, *Annals of Biomedical Engineering*, Vol 25, 975-984
12. Ropella, K. et al, “Differentiation of ventricular tachyarrhythmias”, 1990, *Circulation*, Vol 82, 2035-2043
13. Ash, R., *Information Theory*. New York: Dover Publications Inc., 1990. Print.
14. Mcdonnell, M.D. et al, “An introductory review of information theory in the context of computational neuroscience”, 2011, *Biological Cybernetics*, Vol 105, 55-70, DOI 10.1007/s00422-011-0451-9
15. Shannon, C.E., “A Mathematical Theory of Communication”, 1948, *The Bell System Technical Journal*, Vol 27, 379-423, 623-656
16. Dimitrov, A.G. et al, “Information theory in neuroscience”, 2011, *Journal of Computational Neuroscience*, Vol 30, 1-5, DOI 10.1007/s10827-011-0314-3
17. Jaynes, E.T., “Information Theory and Statistical Mechanics”, 1957, *Physical Review,* Vol 106(4)
18. Paninski, L., “Estimation of Entropy and Mutual Information”, 2003, *Neural Computation*, Vol 15, 1191-1253
19. Rapaport, A. et al, “The theoretical channel capacity of a single neuron as determined by various coding systems”, 1960, *Information and Control*, Vol 3(4), 335-350.

20. Quinn, C.J. et al, "Estimating the directed information to infer causal relationships in ensemble neural spike train recordings", 2011, *Journal of Computational Neuroscience*, doi:10.007/s10827-010-0247-2

21. Brodmann, K., "Brodmann's localisation in the Cerebral Cortex: The Principles of Comparative Localisation in the Cerebral Cortex Based on Cytoarchitectonics", 1909, translated by Laurence J. Garey (2006)

22. Rubinov, M. et al, "Complex network measures of brain connectivity: Uses and interpretations", 2010, *NeuroImage*, Vol 52, 1059-1069

23. Boccaletti, S. et al, "Complex networks: Structure and dynamics", 2006, *Physics Reports*, Vol 424, 175-308

24. University of Michigan Department of Cognitive Neuroscience, "Brodmann", <http://umich.edu/~cogneuro/jpg/Brodmann.html>

25. Fan, L. et al, "The Human Brainnetome Atlas: A New Brain Atlas Based on Connectional Architecture", 2016, *Cerebral Cortex*, doi: 10.1093/cercor/bhw157

26. Wang, J. et al, "A Novel Segmentation, Mutual Information Network Framework for EEG Analysis of Motor Tasks", 2009, *BioMedical Engineering OnLine*, pp. 8:9, doi:10.1186/1475-925X-8-9

27. Jeong, J. et al, "Mutual information analysis of the EEG in patients with Alzheimer's disease", 2001, *Clinical Neurophysiology*, Vol 112, pp. 827-835

28. Lee, P, "Mutual Information Derived Functional Connectivity of the Electroencephalogram (EEG)", 2007, (Unpublished master's thesis). University of British Columbia, Vancouver, BC, Canada

29. Vicente, R. et al, “Transfer entropy—a model-free measure of effective connectivity for the neurosciences”, 2011, *Journal of Computational Neuroscience*, Vol 30, 45-67, DOI 10.1007/s10827-010-0262-3

30. Razak, F.A. et al, “Quantifying ‘Causality’ in Complex Systems: Understanding Transfer Entropy, 2014, *PLoS ONE* 9(6): e99462, doi: 10.1371/journal.pone.0099462

31. Darbellay, G.A. et al, “Estimation of the Information by an Adaptive Partitioning of the Observation Space”, 1999, *IEEE Transactions on Information Theory*, Vol 45

32. Lee, J. et al, “Transfer Entropy Estimation and Directional Coupling Change Detection in Biomedical Time Series”, 2012, *BioMedical Engineering OnLine*, 11-19, doi:10.1186/1475-925X-11-19

33. Le, T.K., “Information dependency: Strong consistency of Darbellay-Vajda partition estimators”, 2013, *Journal of Statistical Planning and Inference*, Vol 143, 2089-2100

34. Dosenbach, N.U.F. et al, “Distinct brain networks for adaptive and stable task control in humans”, 2007, *Proceedings of the National Academy of Sciences*, Vol 104(26), 11073-11078

35. Fair, D.A. et al, “Development of distinct control networks through segregation and integration”, 2007, *Proceedings of the National Academy of Sciences*, Vol 104(33), 13507-13512

36. Staines, R.W. et al, “Frontal-parietal event-related potential changes associated with practicing a novel visuomotor task”, 2002, *Cognitive Brain Research*, Vol 13, 195-202

37. Tetsuya, I. et al, “Frontoparietal Network Involved in Successful Retrieval from Episodic Memory. Spatial and Temporal Analyses Using fMRI and ERP”, 2006, *Cerebral Cortex*, Vol 16, 1349-1360

38. Sauseng, P. et al, “Fronto-parietal EEG coherence in theta and upper alpha reflect central executive functions of working memory”, 2005, *International Journal of Psychophysiology*, Vol 57, 97-103

39. Orekhova, E.V. et al, “EEG theta rhythm in infants and preschool children”, 2006, *Clinical Neurophysiology*, Vol 117, 1047-1062

40. Klimesch, W., “EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis”, 1998, *Brain Research Reviews*, Vol 29, 169-195

41. Hagemann, D. et al, “The effects of ocular artifacts on (lateralized) broadband power in the EEG”, 2001, *Clinical Neurophysiology*, Vol 112(2), 215-231

42. Sauseng, P. et al, “What does phase information of oscillatory brain activity tell us about cognitive processes?”, 2008, *Neuroscience and Biobehavioral Reviews*, Vol 32, 1001-1013

43. Duncan, J. et al, “Common regions of the human frontal love recruited by diverse cognitive demands”, 2000, *Trends in Neuroscience*, Vol 23, 475-483

44. Chayer, C. et al, “Frontal Lobe Functions”, 2001, *Current Neurology and Neuroscience Reports*, Vol 1, 547-552

45. Fogassi, L. et al, “Motor functions of the parietal lobe”, 2005, *Neurobiology,* Vol 15, 626-631

46. Wagner, A.D. et al, “Parietal lobe contributions to episodic memory retrieval”, 2005, *TRENDS in Cognitive Sciences*, Vol 9, No. 9

47. Husain, M. et al, “Space and the parietal cortex”, 2006, *TRENDS in Cognitive Sciences*, Vol 11, No. 1

48. Kakizaki, T., “Occipital midline EEG and subjective rating of task difficulty as indices of mental task strain”, 1987, *European Journal of Applied Physiology*, Vol 56, 163-168

49. Freeman, W.J. et al, *Imaging Brain Function With EEG: Advanced Temporal and Spatial Analysis of Electroencephalographic Signals*. New York, NY: Springer New York, 2013, Print.

50. Adolph, K.E. et al, “Developmental Continuity? Crawling, Cruising, and Walking”, 2012, *Developmental Science*, Vol 14(2), 306-318

51. Palisano, R. et al, “Development and reliability of a system to classify gross motor function in children with cerebral palsy”, 1997, *Developmental Medicine & Child Neurology*, Vol 39(2), 214-223

52. Soleymani, Z. et al, “The Communication Function Classification System: Cultural adaptation, validity, and reliability of the Farsi version for patients with cerebral palsy”, 2015, *Pediatric Neurology*, Vol 52(3), 333-337

53. Sellers, D. et al, “Development and reliability of a system to classify the eating and drinking ability of people with cerebral palsy”, 2014, *Developmental Medicine & Child Neurology*, Vol 56(3), 245-251

54. Backman, C.L. et al, “Single-Subject Research in Rehabilitation: A Review of Studies Using AB, Withdrawal, Multiple Baseline, and Alternating Treatment Designs”, 1997, *Archives of Physical Medicine and Rehabilitation*, Vol 78(10), 1145-1153

55. Cellucci, C.J. et al, “Statistical validation of mutual information calculations: Comparison of alternative numerical algorithms”, 2005, *Physical Review*, Vol 71(6), 066208(14)

56. Cooper, P.S. et al, “Theta frontoparietal connectivity associated with proactive and reactive cognitive control processes”, 2014, *NeuroImage*, Vol 108, 354-363

57. Babiloni, C. et al, “Anticipation of somatosensory and motor events increases centro-parietal functional coupling: An EEG coherence study”, 2005, *Clinical Neurophysiology*, Vol 117(5), 1000-1008

58. Sauseng, P. et al, “Two brakes are better than one: The neural bases of inhibitory control of motor memory traces”, 2013, *NeuroImage*, Vol 65, 52-58

59. Wiener, N., “The theory of prediction”, 1956, *Modern mathematics for engineers*, 1, 125-139, New York: McGraw-Hill. Print.

60. Butler, C. et al, “Motorized wheelchair driving by disabled children”, 1984, *Physical Medicine and Rehabilitation*, Vol 65(2), 95-97

61. Nilsson, L. et al, “Assessment of learning powered mobility use- Applying grounded theory to occupational performance”, 2014, *Journal of Rehabilitation Research and Development*, Vol 51(6), 963-974

62. Law, M. et al, “Canadian occupational performance measure”, 1998, Ottawa: Canadian Association of Occupational Therapists. Print.

63. Bodiam, C., “The Use of the Canadian Occupational Performance Measure for the Assessment of Outcome on a Neurorehabilitation Unit”, *British Journal of Occupational Therapy*, 1999, Vol 62(3), 123-126

64. Pineda, J.A., “The functional significance of mu rhythms: Translating “seeing” and “hearing” into “doing””, 2005, *Brain Research Reviews*, Vol 50(1), 57-68

65. Pfurtschellar, G. et al, “Functional dissociation of lower and upper frequency mu rhythms in relation to voluntary limb movement”, 2000, *Clinical Neurophysiology*, Vol 111(10), 1873-1879

66. Perry, A. et al, “Intranasal oxytocin modulates EEG mu/alpha and beta rhythms during perception of biological motion”, 2010, *Psychoneuroendocrinology*, Vol 35(10), 1446-1453

67. Wong, K.F. et al, “Robust Time-Varying Multivariate Coherence Estimation: Application to Electroencephalogram Recordings During General Anesthesia”, 2011, *Conf Proc IEEE Eng Med Biol Soc.*, 4725-4728