

**LARGE SCALE FUNCTIONAL CONNECTIVITY NETWORKS OF RESTING STATE  
MAGNETOENCEPHALOGRAPHY**

by

**Benjamin T. Schmidt**

BS in Bioengineering, University of Pittsburgh, 2008

Submitted to the Graduate Faculty of  
Swanson School of Engineering in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy

University of Pittsburgh

2017

UNIVERSITY OF PITTSBURGH  
SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Benjamin T. Schmidt

It was defended on

April 3, 2017

and approved by

Tamer S. Ibrahim, Ph.D., Associate Professor, Department of Bioengineering

Avniel S. Ghuman, Ph.D., Assistant Professor, Department of Neurological Surgery

George D. Stetten, M.D., Ph.D., Professor, Department of Bioengineering and Carnegie Mellon University Robotics Institute

Dissertation Director: Theodore J. Huppert, Ph.D., Associate Professor, Departments of Bioengineering and Radiology

Copyright © by Benjamin T. Schmidt

2017

# **LARGE SCALE FUNCTIONAL CONNECTIVITY NETWORKS OF RESTING STATE MAGNETOENCEPHALOGRAPHY**

Benjamin T. Schmidt, PhD

University of Pittsburgh, 2017

Understanding relationships between cortical neural activity is an important area of research. Investigations of the neural dynamics associated with healthy and disordered brains could lead to new insights about disease models. Functional connectivity is a pattern of statistical activity and is a promising method for investigating these neural dynamics by observing intrinsic neural activity arising during spontaneous cortical activations recorded via magnetoencephalography (MEG). MEG is a non-invasive measure of the magnetic fields produced during neural activity and provides information regarding neural synchrony.

Phase locking is a time frequency analysis method that provides frequency band-specific results of neural communication. Leveraging multiple computers operating in a cluster extends the scale of these investigations to whole brain functional connectivity. Quantification of these large-scale networks would allow for the characterization of connectivity in a mathematically rigorous manner.

However, the volume of data required to characterize these networks creates a multiple comparison problem (MCP) in which upward of 33 million simultaneous hypotheses are tested. Conservative approaches such as Bonferroni can result in the loss of statistical power while more

liberal methods may under-correct, therefore leading to an increase in the true type I error rate. In this work, we used a combination of functionally defined cortical surface clustering methods followed by a non-parametric permutation testing paradigm to control the family-wise error rate and provide statistically valid networks.

These methods were validated with simulation studies to characterize limitations in inferences from the resultant whole brain networks. We then examined MEG of healthy subjects during resting state recordings to characterize intrinsic network activity across four physiological frequency bands.

Quantifying large-scale functional connectivity networks allowed for the investigation of electrophysiological networks within specific frequency bands. The understanding of intrinsic network connections allows for better understanding of the electrophysiological processes underlying brain function. While the current dissertation is restricted to healthy brains, the resulting quantification of these networks allows future studies to explore the ability of network aberrations to predict disordered brain states.

## TABLE OF CONTENTS

<b>1.0</b>	<b>INTRODUCTION .....</b>	<b>1</b>
<b>1.1</b>	<b>MOTIVATION AND SIGNIFICANCE .....</b>	<b>2</b>
<b>1.2</b>	<b>RESEARCH OBJECTIVES.....</b>	<b>5</b>
<b>1.3</b>	<b>DISSERTATION STRUCTURE .....</b>	<b>6</b>
<b>1.3.1</b>	<b>Chapter 2 .....</b>	<b>6</b>
<b>1.3.2</b>	<b>Chapter 3 .....</b>	<b>6</b>
<b>1.3.3</b>	<b>Chapter 4 .....</b>	<b>6</b>
<b>1.3.4</b>	<b>Chapter 5 .....</b>	<b>7</b>
<b>1.3.5</b>	<b>Chapter 6 .....</b>	<b>7</b>
<b>1.3.6</b>	<b>Chapter 7 .....</b>	<b>7</b>
<b>1.3.8</b>	<b>Chapter 8 .....</b>	<b>8</b>
<b>2.0</b>	<b>FUNCTIONAL CONNECTIVITY: PHASE LOCKING.....</b>	<b>9</b>
<b>2.1</b>	<b>INTRODUCTION .....</b>	<b>9</b>
<b>2.2</b>	<b>BACKGROUND .....</b>	<b>10</b>
<b>2.2.1</b>	<b>Anatomical Connectivity .....</b>	<b>10</b>
<b>2.2.2</b>	<b>Functional Connectivity .....</b>	<b>12</b>
<b>2.3</b>	<b>PHASE LOCKING VALUE.....</b>	<b>15</b>
<b>2.3.1</b>	<b>Phase Locking Value.....</b>	<b>16</b>
<b>2.4</b>	<b>INSTANTEOUS PHASE .....</b>	<b>18</b>

2.4.1	Wavelets .....	19
2.4.2	Morlet Wavelet.....	19
2.4.3	Explicit Parameterization of Morlet Wavelet by Frequency .....	23
2.4.4	Analytic Signal and Hilbert Transform.....	26
2.4.5	Numerical Comparison of Hilbert and Morlet Methods in PLV .....	26
2.6	PHASE LOCKING VALUE STATISTICS .....	32
2.6.2	Calculation of PLV with Known Phase Distributions.....	37
2.6.3	Uniform Distribution of Phase Differences .....	37
2.6.4	Phase Difference of Two Uniform Distributions.....	40
2.7	RELATIONSHIP BETWEEN PLV AND CORRELATION.....	45
2.9	SUMMARY .....	57
2.9.1	Time-Frequency Resolution: Implications On Phase Locking.....	58
2.11	CONCLUSION .....	65
3.0	PHASE LOCKING GRAPHS.....	66
3.1	INTRODUCTION .....	66
3.2	BACKGROUND .....	69
3.3	METHODS.....	72
3.3.1	Network Centrality .....	72
3.3.2	Functionally Defined Seed Regions .....	75
3.4	NETWORK ANALYSIS.....	82
3.4.1	Large Scale Statistical Testing.....	82
3.4.2	Cluster Network Statistics.....	88
3.5	CONCLUSION .....	93
4.0	LEGION: DISTRIBUTED COMPUTING.....	94
4.1	INTRODUCTION .....	94

4.1.2	Distributed Computing.....	96
4.1.3	Computing Environment.....	98
4.2	<b>SOFTWARE ARCHITECTURE.....</b>	<b>99</b>
4.2.1	Kernel Functions.....	99
4.2.3	Processing Components - Grid .....	101
4.2.5	Master .....	102
4.2.6	Jobrunner .....	103
4.3	<b>STREAMING API.....</b>	<b>103</b>
4.3.1	Optimization and Customization Of Streaming API .....	104
4.3.2	Input Pair-Wise Execution.....	105
4.5	<b>DISTRIBUTED IMPLEMENTATION OF PHASE LOCKING .....</b>	<b>109</b>
4.7	<b>CONCLUSION .....</b>	<b>111</b>
4.7.1	Caveats and Limitations.....	111
4.7.3	Future Work.....	113
5.0	<b>WHOLE BRAIN PHASE LOCKING GRAPHS IN MAGNETOENCEPHALOGRAPHY .....</b>	<b>114</b>
5.1	<b>INTRODUCTION .....</b>	<b>114</b>
5.2	<b>BACKGROUND.....</b>	<b>115</b>
5.3	<b>METHODS.....</b>	<b>120</b>
5.3.1	Cortical Surface Reconstruction .....	120
5.3.2	Minimum Norm Estimation of Cortical Locations.....	120
5.3.3	Properties of MEG Cortical Reconstruction.....	124
5.3.4	MEG Simulation .....	127
5.3.5	Phase Locking Graphs in MEG.....	129
5.4	<b>RESULTS.....</b>	<b>134</b>
5.4.1	Empty Room Noise .....	134

5.4.2	Simulated Point Generator Phase Locking Properties.....	138
5.4.3	Phase Locking Resolution .....	145
5.4.4	Effects of Signal to Noise Ratio on Phase Locking I.....	148
5.4.6	Positive Control.....	150
5.4.8	Negative Control .....	154
5.5	DISCUSSION AND CONCLUSION .....	157
6.0	APPLICATION TO WHOLE BRAIN PHASE LOCKING: EXPERIMENTAL DATA.....	160
6.1	INTRODUCTION AND BACKGROUND .....	160
6.2	METHODS.....	161
6.2.1	Resting State Data Collection .....	161
6.2.2	Subjects .....	162
6.2.4	Subject registration.....	163
6.2.5	Phase Locking Analysis .....	166
6.3	RESULTS.....	169
6.3.1	Centrality Measures.....	170
6.3.2	Theta Band .....	178
6.3.3	Alpha Band.....	184
6.3.4	Beta-Low Band.....	189
6.3.5	Beta-High Band.....	193
6.3.6	Inter-Band Results .....	197
6.4	DISCUSSION.....	199
6.5	CONCLUSION .....	201
7.0	WHOLE BRAIN FUNCTIONAL CONNECTIVITY USING DATA-DRIVEN PHASE LOCKING MEASURES OF RESTING STATE MEG.....	203
7.1	INTRODUCTION .....	203

<b>7.3</b>	<b>MATERIALS AND METHODS</b> .....	<b>207</b>
7.3.1	Resting State Phase Locking.....	209
7.3.2	Dipole Centrality .....	212
7.3.3	Anatomically biased functionally defined regions .....	213
7.3.4	Statistical testing of regional connections.....	217
7.3.5	Description of Numerical Simulations .....	218
7.3.6	Evaluation of PLV Characteristics.....	222
<b>7.4</b>	<b>EXPERIMENTAL METHODS</b> .....	<b>224</b>
7.4.1	Subject population .....	224
7.4.2	MEG recording and preprocessing .....	224
7.4.3	Structural MRI and MEG Source localization .....	225
<b>7.5</b>	<b>RESULTS</b> .....	<b>227</b>
7.5.1	Numerical Simulations .....	227
7.5.2	Performance of data-driven methods.....	231
7.5.3	Characterization of errors in cluster spatial location.....	233
7.5.4	PLV characteristics.....	234
7.5.5	Point spread function limitation .....	237
7.5.6	Evaluation of statistical methods.....	237
<b>7.6</b>	<b>EXPERIMENTAL RESULTS</b> .....	<b>238</b>
<b>8.0</b>	<b>DISCUSSION</b> .....	<b>242</b>
<b>8.1</b>	<b>LIMITATIONS</b> .....	<b>244</b>
<b>8.3</b>	<b>FUTURE DIRECTIONS</b> .....	<b>247</b>
<b>APPENDIX A</b> .....		<b>248</b>
<b>APPENDIX B</b> .....		<b>259</b>
<b>BIBLIOGRAPHY</b> .....		<b>262</b>

## LIST OF FIGURES

Figure 2-1 Identification of functionally connected regions.....	13
Figure 2-2 An example of functional connectivity derived networks .....	14
Figure 2-3 Morlet wavelet at multiple scales in the frequency domain.....	22
Figure 2-4 Temporal and Spectral Scaling of Morlet Wavelet.....	25
Figure 2-5 PLV values using Hilbert and Morlet methods in Non-Narrow Band Model .....	28
Figure 2-6 PLV Values between Hilbert and Morlet in Narrow-Band Model .....	30
Figure 2-7 PLV for expanding uniform distributions.....	39
Figure 2-8 A generic boxcar function.....	41
Figure 2-9 PLV as a function of width parameter of two boxcar phase distributions .....	42
Figure 2-10 Result of convolving two identical independent boxcar functions .....	43
Figure 2-11 Probability distribution of phase angles.....	44
Figure 2-12 Results of wavelet parameters on temporal and spectral bandwidths.....	58
Figure 2-13 Average phase locking value as a function of frequency and wavelet parameter ....	59
Figure 2-14 Time-Frequency resolution dependence upon wavelet frequency bandwidth .....	61
Figure 2-15 Resolution dependence upon wavelet frequency bandwidth with noise.....	63
Figure 3-1 Identification of Functionally Defined Regions.....	68
Figure 3-2 Graph Representation.....	70
Figure 3-3 Graph Centrality.....	73
Figure 3-4 Eigenvector Centrality Real Phase Locking Graphs.....	74

Figure 3-5 Cosine distance.....	79
Figure 3-6 Spherical Clustering.....	81
Figure 3-7 Permutation testing on Clusters. ....	90
Figure 3-8 A representative network identified in subject population .....	92
Figure 4-1 Legion Architecture. ....	100
Figure 4-2 Legion Pairwise Computation.....	105
Figure 4-3 Block distribution mechanisms for distributed computation. ....	107
Figure 5-1 MEG System. ....	117
Figure 5-2 Cortical Surface Dipole Locations.....	121
Figure 5-3 Vertex Numbering After Cortical Reconstruction.....	125
Figure 5-4 Average distances from vertices as a result of vertex numbering pattern.....	126
Figure 5-5 MEG Phase Locking Graphs.....	130
Figure 5-6 Phase Locking Adjacency Matrix. ....	132
Figure 5-7 Representation of Phase Locking Graph.....	133
Figure 5-8 Empty Room Noise Example.....	135
Figure 5-9 Visualization of empty room error at base of brain. ....	137
Figure 5-10 Effect of sinusoidal generator on point spread function. ....	138
Figure 5-11 Two-point generators with overlapping phase locking. ....	140
Figure 5-12 Point spread function of two point generators in one-dimensional vertex indices. ....	141
Figure 5-13 Three point simulation. ....	144
Figure 5-14 AB Simulation vertices. ....	145
Figure 5-15 Point spread function overlap.....	147
Figure 5-16 Source signal power and relation to false positive spatial correlations. ....	148

Figure 5-17 Phase locking false positives as function of distance.....	149
Figure 5-18 Positive Control Seed Regions.....	150
Figure 5-19 Positive Control Results.....	151
Figure 5-20 Significant Cluster Phase Locking Connectivity. ....	153
Figure 5-21 Negative Control Data Creation.....	155
Figure 5-22 Negative Control Results. ....	156
Figure 6-1 Cortical surface registration to a common atlas.....	165
Figure 6-2 Averaging of Phase Locking Bands.....	166
Figure 6-3 Phase Locking Permutation Testing Between Subjects.....	168
Figure 6-4 Subject Specific Eigenvector Centrality of Resting State Phase Locking.....	172
Figure 6-5 Theta Band Network of EVC.....	173
Figure 6-6 Alpha Network of EVC.....	174
Figure 6-7 Beta-Low Network of EVC. ....	175
Figure 6-8 Beta-High Network of EVC.....	176
Figure 6-9 Mean and Standard Deviation of EVC Across All Frequencies.....	177
Figure 6-10 Theta Band Adjacency Matrix. ....	180
Figure 6-11 Theta Band Intra-hemisphere Networks. ....	181
Figure 6-12 Theta Band Inter-hemisphere Networks. ....	183
Figure 6-13 Alpha Band Adjacency Matrix.....	185
Figure 6-14 Alpha Band Intra-hemisphere Networks.....	186
Figure 6-15 Alpha Band Inter-hemisphere Networks.....	187
Figure 6-16 Alpha Band Temporal Symmetry. ....	188
Figure 6-17 Beta-Low Band Adjacency Matrix. ....	190

Figure 6-18 Beta-Low Band Intra-hemisphere Networks. ....	191
Figure 6-19 Beta-Low Band Inter-hemisphere Networks. ....	192
Figure 6-20 Beta-High Band Adjacency Matrix.....	194
Figure 6-21 Beta-High Band Intra-hemisphere Networks.....	195
Figure 6-22 Beta-High Band Inter-hemisphere Networks.....	196
Figure 6-23 Alpha/Theta Intra-hemisphere Networks.....	198
Figure 6-24 Beta-Low/Beta-High Intra-Hemisphere Networks. ....	198
Figure 7-1 Overview of Method .....	207
Figure 7-2 Dendrogram.....	216
Figure 7-3 Calculation of Full Width Half Max (FWHM).....	221
Figure 7-4 Identification of functionally defined regions in subject data.....	227
Figure 7-5 TPR of FWHM and Dipole containing clusters.....	229
Figure 7-6 ROC curves for brain simulation with area under curve.....	230
Figure 7-7 FWHM centroid location in simulation as a function of inter-dipole distance.....	233
Figure 7-8 ROC Curves of simulation.....	236
Figure 7-9 FWHM centroid location in simulation as a function of inter-dipole distance.....	239
Figure 7-10 Graph of alpha band network clusters over the arcuate fasciculus .....	240

## 1.0 INTRODUCTION

Understanding relationships between cortical neural activities is important for investigation of the neural dynamics associated with the healthy and disordered brain. Functional connectivity is a pattern of statistical activity and is a promising method for investigating these neural dynamics. Intrinsic neural activity arising during spontaneous cortical activations correlates with patterns of neurophysiological signaling that arise during the presentation of stimuli to subjects.

Phase locking is a functional connectivity method that quantifies the variability between time series by analyzing point-by-point differences in instantaneous phases. It utilizes a time frequency analysis of the time series that provides frequency band specific results. Previous work has shown that functional networks communicate at specific frequencies. Therefore, phase locking methods can be applied to functional connectivity to provide for direct investigation of these frequency-specific networks.

Technological limitations from the computationally intensive phase locking methods in neurophysiological data create difficulties in analyzing large-scale network interactions. By leveraging multiple computers operating in a cluster, the previously computationally intractable problem of measuring whole brain functional connectivity becomes possible.

The application of large-scale functional connectivity at specified frequency bands provides new insights into the intrinsic functioning of cortical networks. Quantification of these networks allows investigation of frequency band specific network characteristics. Further,

application of these methods to whole brain networks can elucidate connections across the entire cortex. Eventually, the ability to accurately quantify these networks could lead to a better understanding of how aberrations in network connectivity relate to disease models.

## **1.1 MOTIVATION AND SIGNIFICANCE**

Functional connectivity is a promising method for better understanding the neural dynamics of large-scale cortical activity [1, 2]. However, the computational size of the problem can sometimes overwhelm the experimental and analytic methods of processing the recorded data. This has necessitated that many research studies restrict their focus to specific hypothesis-driven questions by using regions of interest (ROIs) within the full brain space. However, in the context of improved computational power, it is possible to divide repeated but independent calculations can parallelize many functional connectivity methods across clusters of computers.

By leveraging computer clusters it is possible to provide novel neuroscientific methods to access problems whose analysis would easily overwhelm single computers. However, the primary detractor to large-scale computation, besides computational complexity, is the multiple comparison problem (MCP). The MCP arises from increased type I errors resulting from multiple simultaneous statistical tests. Conservative approaches such as Bonferroni can eliminate all results while more liberal methods may under-correct leading to an increase in the true type I error rate compared to the prediction. To address this, we use a combination of functionally identified surface clustering followed by non-parametric methods to establish the global type I error and provide robust statistical networks.

There has been interest in previous literature towards using Graph Theory to represent cortical networks [3, 4]. Graphs provide a natural framework for understanding and quantifying networks. Graphs can be represented as a set of vertices that are connected to one another via edges. This work builds and expands on those previous efforts by demonstrating methods by which graph theory frameworks, such as centrality, can be used in conjunction with imaging modalities to investigate networks in multiple time series signals. In doing so, we establish methods by which future work can conduct quantitative analyses of these complex data structures.

Neuroimaging modalities that have been previously employed in functional connectivity studies fall roughly into hemodynamic and electrophysiological categories. Functional magnetic resonance imaging methods record changes in cerebral hemodynamic activity as the result of neural activation and are typically recorded at a .5 – 1 Hz sampling rate [5]. Electroencephalography (EEG) is a non-invasive electrophysiological study that measures voltage fluctuations resulting from ionic currents in the brain and is typically recorded at a sampling rate between 200 Hz and 2k Hz [6]. An alternative electrophysiological method, magnetoencephalography (MEG) records magnetic fields produced by electric currents in the brain and is typically recorded with a sampling rate of 1 kHz [2]. In this dissertation, we have confined our analysis methods to electrophysiological studies and more specifically to MEG. However, these methods could be adapted to any time series neuroimaging modality and could therefore be used in fMRI or EEG. MEG was chosen for this dissertation because of its high sample resolution and spatial coverage. However, fMRI or EEG could also be used with some modification to the analysis methods.

Quantifying large-scale functional connectivity networks allows for the investigation of healthy and disordered brains. Understanding the intrinsic network connections can improve our understanding of the electrophysiological processes underlying brain function. Quantification of these networks would also allow future studies to explore the ability of network aberrations to predict disordered brain states.

## 1.2 RESEARCH OBJECTIVES

The primary objective of this research dissertation is the development of methods to compute statistically valid whole brain MEG phase locking networks and the application of those methods to resting state MEG. Properties of phase locking as they relate to whole brain MEG will be investigated for their effects upon the resultant networks. Statistical tests that are robust to the large multiple comparison problems inherent in neuroimaging datasets will also be investigated to control the rate of type I errors. Application of graph methods will allow whole brain networks to be analyzed in a robust mathematical sense. Neurophysiological interpretations of these network graphs will allow cortical neural networks to be defined. Validation simulations will be conducted to ensure reliability of networks established using these methods. Finally, application of phase locking graphs will be applied to resting state MEG scans. Significant networks in specific frequency bands will be elucidated.

## **1.3 DISSERTATION STRUCTURE**

### **1.3.1 Chapter 2**

This chapter discusses the background of neuroanatomical functional connectivity studies. We then introduce the reader to phase locking, a frequency-dependent functional connectivity analysis method. Theoretical and practical considerations of the usage of phase locking in functional neuroanatomy studies are also investigated.

### **1.3.2 Chapter 3**

This chapter discusses the creation of a framework for the analysis of functional connectivity estimates in large-scale neural recordings. Phase locking was calculated between large numbers of neural populations yielding phase locking graphs. Methods to test for statistically significant networks were established that considered the implication of multiple statistical inferences in assessing statistically significant networks.

### **1.3.3 Chapter 4**

We describe a custom software library, Legion, that was written to perform distributed parallel computation of functional connectivity estimates in a time-efficient manner. Phase locking is a computationally intensive algorithm and when applied to large neuroimaging data sets it quickly scales to become computationally intractable. By utilizing clusters of computers that execute parts of the problem in parallel, the computational speed is greatly improved.

### **1.3.4 Chapter 5**

The previously established methods of phase locking graphs are applied to magnetoencephalography. Simulation experiments are conducted to characterize the results of the network analysis methods given *a priori* neural dynamics. In addition, investigations of the empty room phase locking graph are conducted to validate our methods with respect to the characteristics of MEG reconstruction.

### **1.3.5 Chapter 6**

Phase locking graph methodologies are applied to resting state MEG data of healthy human subjects. Statistically significant networks are detected in intra- and inter- hemispheric connections and are interpreted within the context of previous literature regarding frequency-specific network communication in functional connectivity studies.

### **1.3.6 Chapter 7**

This chapter provides a summary of the experimental and numerical simulations utilized in this dissertation. It provides a demonstration of the capabilities of the methods outlined herein as well as a numerical simulation that demonstrate the methods' resolution.

### **1.3.7 Chapter 8**

This chapter discusses conclusions of the work, limitations, and possible future directions.

## **2.0 FUNCTIONAL CONNECTIVITY: PHASE LOCKING**

This chapter discusses the background of neuroanatomical functional connectivity studies. We then introduce the reader to phase locking, a frequency-dependent functional connectivity analysis method. Theoretical and practical considerations of the usage of phase locking in functional neuroanatomy studies are also investigated.

### **2.1 INTRODUCTION**

Functional connectivity is a pattern of statistical associations between distinct units of the nervous system [1, 7, 8]. These units may refer to either single neurons or populations of neurons consisting of anatomically or functionally distinct brain regions.

Functional connectivity provides a means for understanding intrinsic neural activity as well as how neural populations interact relative to a task. As a statistical method, functional connectivity attempts to detect relationships between two neural units guided by assumptions about the model of the underlying neural system. Many methods exist to study different components of this dependence; these methods rely upon time series estimates of brain activity as recorded by neuroimaging modalities including MEG, fMRI, and EEG [2, 5, 6, 9-11]. Time series methods perform repeated observations of the underlying brain's functioning according to

the physics model employed by the given neuroimaging modality. Here we utilize phase locking value (PLV) as a mechanism for understanding the relationship between neuronal populations at specific neurophysiologically relevant frequencies.

## **2.2 BACKGROUND**

Brain connectivity can be broadly categorized into two distinct application areas: anatomical connectivity and functional connectivity [1]. Anatomical connectivity refers to physical connections that may exist at many levels within the nervous system. In contrast, functional connectivity evaluates the relationship between brain regions without relying on physical connections. Both aspects of connectivity are described below as well as some of the approaches that have been taken towards investigating their respective application areas.

### **2.2.1 Anatomical Connectivity**

Anatomical connectivity refers to the physical network of structural connections linking neurons [7, 8]. At the highest resolution, this refers to a population of individual neurons and a characterization of the distinct network of synapse connections between those neurons. Technological limitations are imposed on our ability to resolve anatomical connections at high resolutions in which there is an implicit tradeoff between resolution and scale. For example, microscopy can examine very small populations of neurons and their local neural connections but is impractical at the scale of the entire brain [12]. However, anatomical connectivity patterns

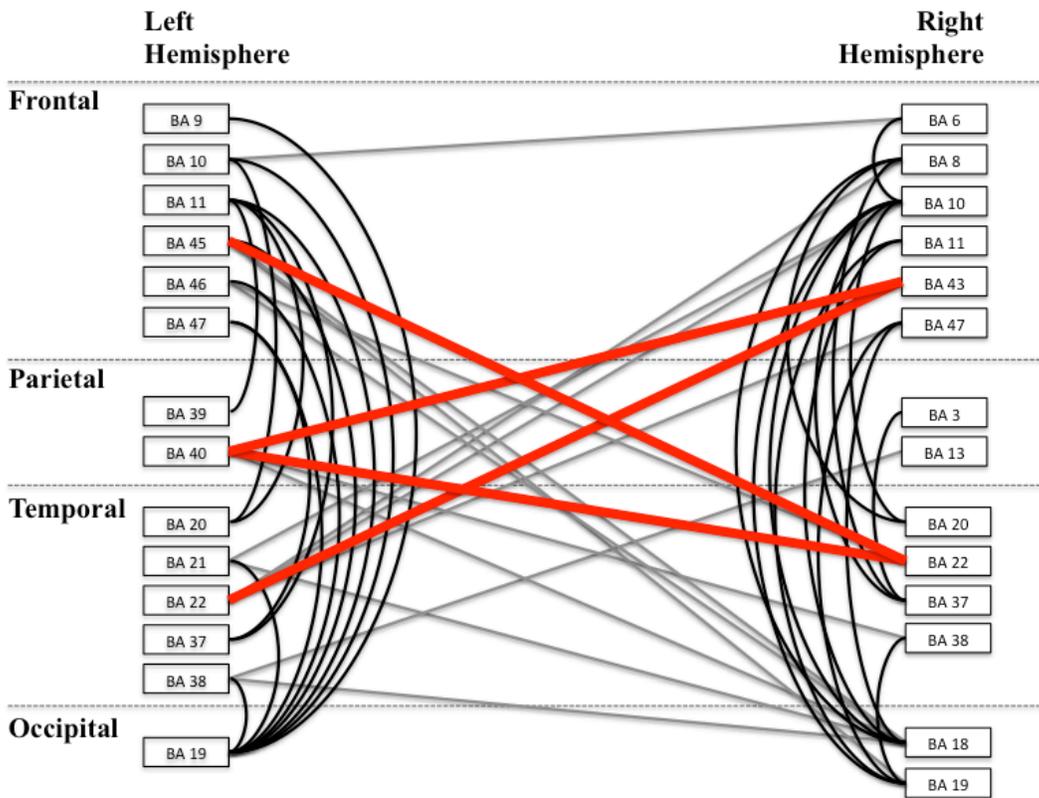
exist at many levels within the nervous systems of complex organisms. At the single neural level they exhibit specific patterns between individual neurons, but they also exhibit distinct patterns at the level of neural populations and the anatomical connectivity between those populations [13, 14]. Regardless of the level of analysis (local neuron connections versus connections between neural populations), technical limitations place a practical limit on the scale and scope of investigation of anatomical connectivity.

The ultimate goal of anatomical connectivity additionally includes the characterization of the exact biophysical properties associated with each neuronal connection. While anatomical connectivity is not directly a time series measurement, a number of factors play a role in the alteration of both the physical connections between neurons and these biophysical parameters. Normal cellular and brain functions such as neural plasticity, cell mitosis, and necrosis can change anatomical connectivity. Therefore, each anatomical map is only a snapshot of the current anatomical connectivity pattern.

At the theoretical apex of anatomical connectivity lies the fully formed connectome [15]. The connectome includes the complete description of every structural element and biophysical property that gives rise to an organism's nervous system. Technological limitations inhibit our ability to record such a map except in the case of organisms like *c. elegans* [16] whose nervous system is relatively simplistic (302 neurons). For more complex nervous systems such as humans, we are limited to observations of large-scale neural populations and the anatomical links between those populations. Methods such as Diffusion Tensor Imaging (DTI) in MRI provide a means for identifying neural tracts via water diffusion in the brain, a method called tractography [17, 18], that can also be useful in understanding varying levels of anatomical connectivity.

### 2.2.2 Functional Connectivity

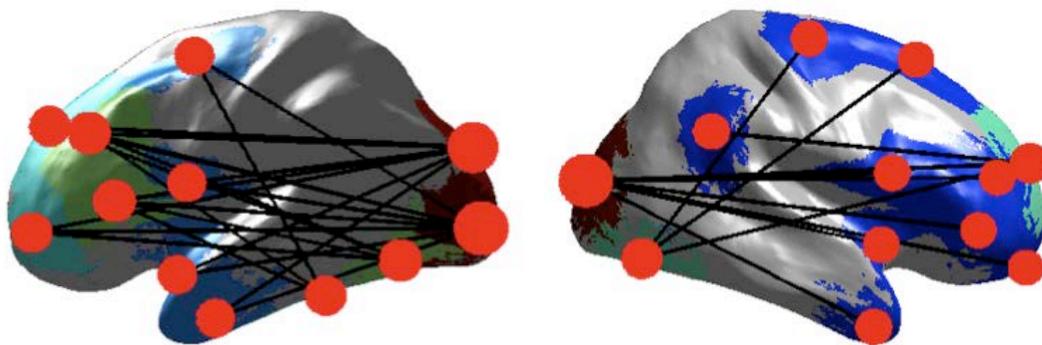
Functional connectivity is different from anatomical connectivity in that it does not rely upon the physical connections between neurons or neuronal populations and instead approaches connectivity as a statistical relationship between neural units. Neural populations can be connected in this manner regardless of the exact physical connection between them. Figure 2-1 shows a functional connectivity map between underlying Brodmann areas that was identified by this methodology (for more information see Chapter 7.0 ). In Figure 2-1, Black lines indicate intra-hemisphere connections, grey lines indicate inter-hemisphere connections, while red is a well-known inter-hemisphere network commonly seen in literature. The ability to identify networks without having to inspect the exact physical connections has tremendous benefits in terms of accessibility and experimental simplicity. These advantages allow us to investigate certain brain characteristics much more easily.



**Figure 2-1 Identification of functionally connected regions**

Functional connectivity is a family of statistical methods used for understanding the interactions between brain regions. Functional connectivity does not rely upon the underlying anatomical connections between brain regions and is a time series method for understanding neural dynamics [1, 15]. Functional connectivity has traditionally been defined in the context of correlation [19], however those definitions do not express the full extent of the possible relationships between time series neural recordings, as we will investigate in this chapter. As such, the addition of other non-correlation based methods yields important insights into the functioning the brain's neural dynamics [20-23]. Below in Figure 2-2, we see an example of a

functional connectivity map overlaid on a brain surface. Functional connectivity techniques allow us to understand how communication works while ignoring the exact method of the underlying physical connections. This has advantages in a structure as complex as the brain and allows us to be able to understand how these networks might vary under various circumstances such as disease, wake-state, or task.



**Figure 2-2 An example of functional connectivity derived networks**

Functional connectivity is driven by mathematical relationships between different brain regions. We are able to define regions of the brain based upon their mathematical relationships. Above can be seen the functionally connected regions (red are nodes, black lines are edges indicating a connection between those underlying regions). With this technique we can begin to understand how different regions of the brain communicate, at what frequencies they communicate, and how that communication might vary under various conditions (disease, sleep, awake etc.)

Many mathematical methods exist for connectivity estimates of time series neural dynamics. In general, any signal processing method for understanding the relationship between two time series can be applied in connectivity studies towards furthering our understanding of how neural units interact. Linear correlation, spectral coherence and Granger's causality have been widely used in literature [20, 24-30], however, we will focus upon phase locking as the primary means of understanding the relationship between neural populations. The following section describes our motivations for selecting phase locking.

### **2.3 PHASE LOCKING VALUE**

Phase locking measures the variability of phase differences between two time series [36]. After spectral decomposition, two time series signals are used as inputs to obtain the instantaneous phase at each time point. Phase differences between the two time series at each point are then averaged and the overall variability of the phase differences is reported. The calculation of phase locking uses time-windowed estimates of spectral composition for the estimation of instantaneous phase. Recent studies suggest that different neural networks communicate at unique frequency bands [34, 35]. As such, the use of spectral domain methods, like phase locking, provides a method for understanding frequency band communications. As such, phase locking may provide additional insights that time-domain analysis methods would not be capable of identifying.

### 2.3.1 Phase Locking Value

We start by looking at how phase locking value (PLV) is obtained beginning with simple sinusoidal signals. We begin with an equation for two time-series signals:

$$x(t) = a_x(t) \cos(w_x t + \theta_x(t)) \quad (2-1)$$

$$y(t) = a_y(t) \cos(w_y t + \theta_y(t)) \quad (2-2)$$

where  $a_x(t)$  is a time-varying amplitude component,  $w_x$  and  $w_y$  are the sinusoidal frequencies, and  $\theta_x(t)$  and  $\theta_y(t)$  are time-varying phases. After application of a time-frequency methodology (i.e. Hilbert transform, or complex wavelet as we will investigate later) we arrive at:

$$z_x(t) = A_x(t) e^{j(w_x t + \phi_x(t))} \quad (2-3)$$

$$z_y(t) = A_y(t) e^{j(w_y t + \phi_y(t))} \quad (2-4)$$

where  $z_x$  and  $z_y$  are the complex analog of the original signals having amplitude components,  $A_x$  and  $A_y$  that are both time-varying and complex and with instantaneous phases of  $\phi_x(t)$  and  $\phi_y(t)$ .

We use expected value notation to arrive at the following equation for PLV:

$$PLV|_{w_x=w_y=w} = |E\{e^{j(\phi_x(t)-\phi_y(t))}\}| \quad (2-5)$$

where the expectation operator is taken as the complex unit vector of the point-wise instantaneous phases. This result is obtained by applying the argument function to the  $z_x$  and  $z_y$  and we arrive at the below equation after assuming both original sinusoidal frequencies are equal.

$$PLV|_{w_x=w_y=w} = |E\{\arg(z_x(t)) - \arg(z_y(t))\}| \quad (2-6)$$

To relate this finding to the finite-sampling used in our experimental results, we introduce the notation for a time-averaged PLV, by using an over-bar. Applying this time-averaging to PLV above, the following equation is obtained:

$$\overline{PLV(w)}|_{w_x=w_y} = \frac{1}{T} \int_0^T |E\{e^{j(\phi_x(t)-\phi_y(t))}\}| dt = \left| E \left\{ \frac{1}{T} \int_0^T e^{j(\phi_x(t)-\phi_y(t))} dt \right\} \right| \quad (2-7)$$

where the integration over a specified time period, T, is applied to the expectation operator using a normalization factor in front. The time-integral can be moved into the expectation operator, in which case only for ergodic and wide sense-stationary (WSS) processes will the PLV value of the series converge to the PLV of the time-sample.

Under assumptions of wide sense stationarity, we can also obtain the PLV equation in discrete time that is more common in literature [3, 28]:

$$P = \frac{1}{N} \left| \sum_{n=1}^N e^{i(\theta_1[n] - \theta_2[n])} \right| \quad (2-8)$$

where  $\emptyset$  has been replaced by  $\theta$  for the two time-series phases.

## 2.4 INSTANTANEOUS PHASE

Our procedure for extracting the instantaneous phase starts from simple sinusoids and extracts a time-series of phases using a suitable time-frequency methodology. In this section we introduce two commonly employed methodologies for obtaining instantaneous phase: complex wavelets and the Hilbert transform.

Within the neuroscientific community, wavelets have become a common methodology for extracting instantaneous phase [2, 38] and the work for this dissertation follows that convention. However, for comparison, we also cover the commonly used alternative methodology of the Hilbert transform for obtaining an analytic signal for use in extracting instantaneous phase.

### **2.4.1 Wavelets**

A wavelet is a brief oscillation waveform that is designed to have certain properties when applied to a time series. As a time-frequency estimation method, wavelets have a specific frequency and temporal resolution that identifies sinusoidal contributions at the given frequency and within the specified time duration. From this complex representation of the original time-series signal, we are able to extract the phase at each time point thus obtaining a representation of the instantaneous phase of the signal.

### **2.4.2 Morlet Wavelet**

Many families of wavelets exist, however the Morlet wavelet has been used extensively in phase locking [2, 38]. The Morlet wavelet is non-orthogonal and has the property of being Gaussian shaped in both the time and frequency domains. Since it is Gaussian shaped, it allows a tradeoff in temporal and spectral width to capture the underlying signal contributions. The Morlet wavelet is complex having both real and imaginary components. The wide-spread use of the Morlet in previous literature, especially those involving neurophysiological signals arising from MEG which we are using in this study, combined with the complex components that allow us to extract the phase components of the signals lead us to use this wavelet [2, 38].

The Morlet wavelet is a complex sinusoid with a Gaussian window envelope and is described by the following equation:

$$\psi(t; \sigma) = \pi^{-\frac{1}{4}} e^{-t^2/2} e^{i\sigma t} \quad (2-9)$$

where  $\sigma$  is the Morlet wavelet parameter. The  $\pi^{-1/4}$  term is a normalization parameter.

The Morlet parameter,  $\sigma$ , can be used to control the resonant frequency of the Morlet wavelet, before application to the signal, by modifying the initial resonant frequency. The first observation in the Morlet wavelet is that the wavelet parameter only appears in the complex exponential term (hence modification of the resonant frequency). Therefore, changes in the wavelet parameter control the underlying oscillation frequency but do not affect the Gaussian envelope (which remains as zero mean and unit standard deviation). Both the Gaussian window and frequency terms can be manipulated to identify different components of the signal by scaling the wavelet.

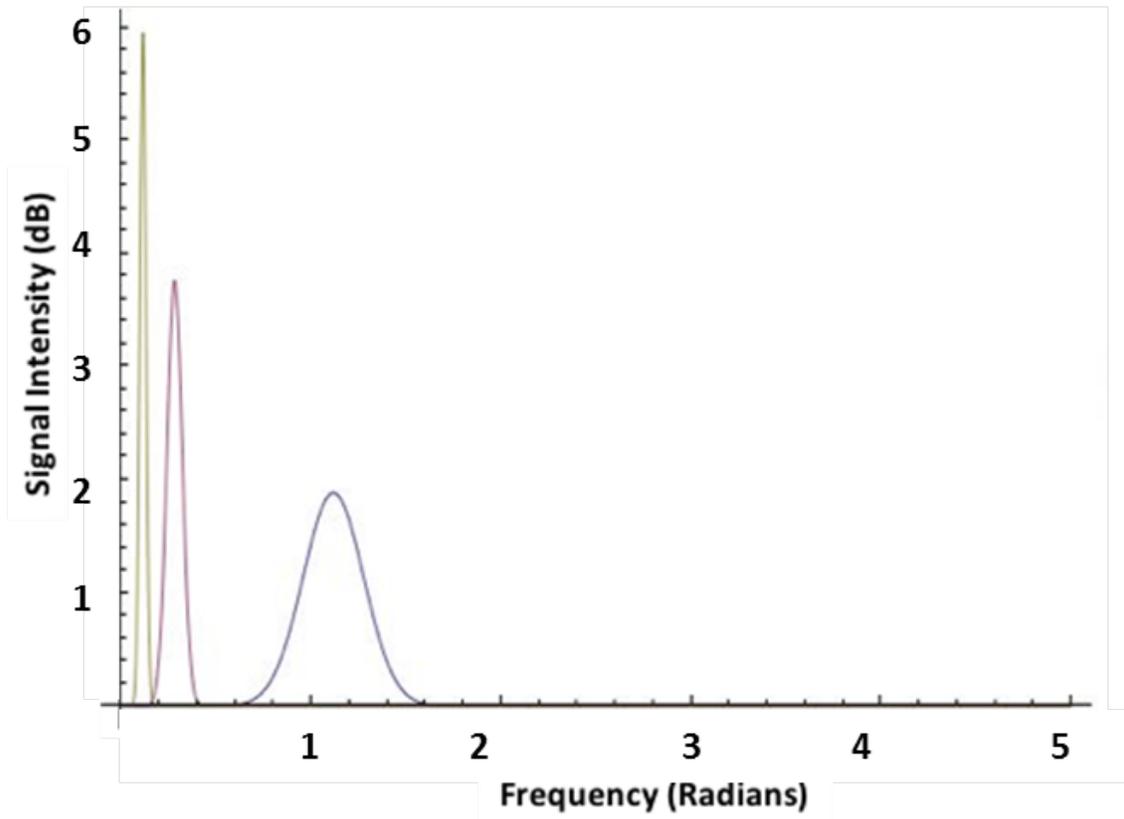
Wavelets can be used to calculate time-frequency properties of a signal by application of the Continuous Wavelet Transform (CWT) of an appropriately scaled wavelet to the original signal to induce resonance at a specific frequency. The scaling parameter is chosen such that the frequencies of interest resonate in the signal under investigation. Applying CWT at multiple time-lagged wavelet locations, allows for calculation of the presence of resonant frequencies as a function of time. The following equation is used for CWT of the original signal at a single wavelet location:

$$X_w(a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} x(t) \psi^* \left( \frac{t-b}{a} \right) dt \quad (2-10)$$

where  $a$  is a scaling parameter and  $a > 0$ ,  $b$  is a time-shift,  $\psi$  is the wavelet function and  $x(t)$  is the signal. The leading term is used to provide normalization for different scaling parameters. By applying the above equation at multiple time shifts,  $b$ , and at multiple scales,  $a$ , a time-frequency representation of the signal is obtained.

Scaling the wavelet changes the frequencies that are resolved. For increasing values of the scaling parameter, the wavelet will resonate at lower frequencies while for decreasing scaling parameters the wavelet resonates at higher frequencies. The above equation can be used to calculate the time-frequency components of the signal at a given frequency.

Scaling the wavelets in the time domain changes the temporal and spectral components at which the wavelet resonates. Increasing the wavelet parameter elongates the wavelet in the temporal domain while in the spectral domain the spectral bandwidth narrows. Accordingly, the narrower spectral bandwidth provides better spectral resolution at the cost of poorer temporal resolution. Conversely, the lower scaling values have wider spectral components that lead to poorer spectral resolution, but in the temporal domain they provide better temporal localization of the signal components. The effects of scaling values can be seen below in Figure 2-3 in which scale = 1 is indicated by blue (farthest right), scale = 4 is indicated by purple (middle), and scale = 10 is indicated by yellow (farthest left)..



**Figure 2-3 Morlet wavelet at multiple scales in the frequency domain**

### 2.4.3 Explicit Parameterization of Morlet Wavelet by Frequency

The Morlet wavelet can be parameterized to explicitly use the standard deviation of the Gaussian term in the temporal and spectral domains and incorporate an explicit parameterization by the center frequency by making the following sequence of substitutions. We start with the equation as above for the Morlet wavelet, but with a scaling parameter:

$$\psi(t; \sigma) = |a|^{-\frac{1}{2}} \pi^{-\frac{1}{4}} e^{-t^2/2a^2} e^{i\sigma t/a} \quad (2-11)$$

The following notational substitution is then made:

$$a = \sigma_t \quad (2-12)$$

This is justified by noting that the  $a$  parameter in the Gaussian term of the original Morlet wavelet is the standard deviation term. The standard deviation in the frequency domain is related to the standard deviation (i.e. the scaling parameter) in the temporal domain by the following:

$$\sigma_f = \frac{1}{2\pi\sigma_t} \quad (2-13)$$

Furthermore, if the wavelet parameter is transformed into frequencies via:

$$\sigma = f_0/\sigma_f \quad (2-14)$$

the following substitution can be made:

$$\sigma = a2\pi f_0 \quad (2-15)$$

After performing these substitutions, the following wavelet form for the Morlet wavelet can be utilized that explicitly uses the bandwidths in the frequency and the temporal domains:

$$\psi(t; f_0) = (\sigma_t\sqrt{\pi})^{-1/2} e^{-t^2/2\sigma_t^2} e^{2i\pi f_0 t} \quad (2-16)$$

where  $\sigma_t$  is the standard deviation of the wavelet in the temporal domain and  $\sigma_f$  is similar for the frequency domain, and  $f_0$  is the frequency of interest. This equation is more frequently seen in the neuroscientific literature than the previous Morlet equation [2, 38]. In our work, we used  $f_0/\sigma_f = 7$  following the previous literature as well as the work by Grossman [39]. The value of 7 allows for an acceptable tradeoff between temporal and spectral resolution with the combination of the neurophysiological frequencies of interest (i.e. 4-50 Hz) as well as the typical recording lengths (~5 minutes at 250 Hz of MEG data).

Unlike the earlier equation for the Morlet wavelet, this form explicitly includes the center frequency parameter rather than the scaling parameter and is useful for calculating specific center frequency versions of the wavelet. The wavelet scaling changes the time-frequency resolution and these changes can be visualized in Figure 2-4 that shows the frequency and

temporal bandwidths as a function of frequency. In Figure 2-4, two standard deviations of the temporal bandwidth (left) and frequency bandwidth (right) of the Morlet wavelet for this studies' wavelet parameter value as a function of frequency.

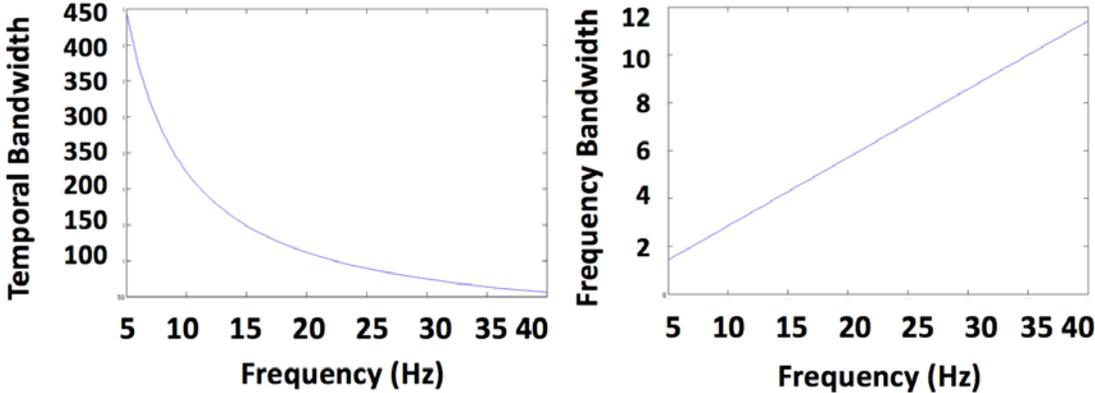


Figure 2-4 Temporal and Spectral Scaling of Morlet Wavelet.

#### 2.4.4 Analytic Signal and Hilbert Transform

Another common methodology for computing instantaneous phase is via the Hilbert transform of a real-valued signal like we have presented here. This Hilbert transform is commonly employed to derive the analytic signal of the original series. It is defined here:

$$H(t) = \int_{-\infty}^{\infty} x(\tau) \frac{1}{\pi(t - \tau)} d\tau \quad (2-17)$$

where  $H(t)$  is the Hilbert transform of the real signal  $x(t)$ . And the analytic signal is defined as:

$$x_a(t) = x(t) + jH(t) \quad (2-18)$$

where  $x_a(t)$  is the analytic signal. And the definition of the instantaneous phase is given by:

$$\phi(t) = \text{arg}[x_a(t)] \quad (2-19)$$

#### 2.4.5 Numerical Comparison of Hilbert and Morlet Methods in PLV

Here we investigated how the use of different time-series methodologies to extract the instantaneous phase affected the final PLV values. Further, we also considered how signals with different levels of correlation impacted the computed PLV.

First, we used the two methods described above, Morlet wavelet or Hilbert transfer, to obtain the instantaneous phase. For the Hilbert transform we used a band pass filter so that the results were a function of a frequency of interest and therefore comparable to the Morlet wavelet method. The band-pass filter used was a 4<sup>th</sup> order Butterworth filter with an 8 Hz width. This was found to produce PLV results comparable to the Morlet within the frequency range of this test (0-40 Hz).

In each test scenario, two signal realizations were produced such that the correlation coefficient was known (both signals were zero mean and unit standard deviation to start). We varied the correlation coefficient between 0 and 1 in steps of 0.1, where 0 is uncorrelated and 1 is perfectly correlated. Two conditions were used, one without a narrow band model and the other with this model. For the non-narrow band model, we utilize the raw time-series data. For the narrow-band model we applied a narrow-band filter (4<sup>th</sup> order Butterworth filter with an 8 Hz width) after the above procedure to obtain correlated signals in a specific band. However, because PLV is not sensitive to amplitude differences, the numerical filtering alone was not found to affect the final PLV value. Therefore, we needed to add random noise to our narrow band model in order to obtain a narrow-band model that provided results in the PLV analysis only at a specific frequency band. We added random noise to obtain an SNR of 1.

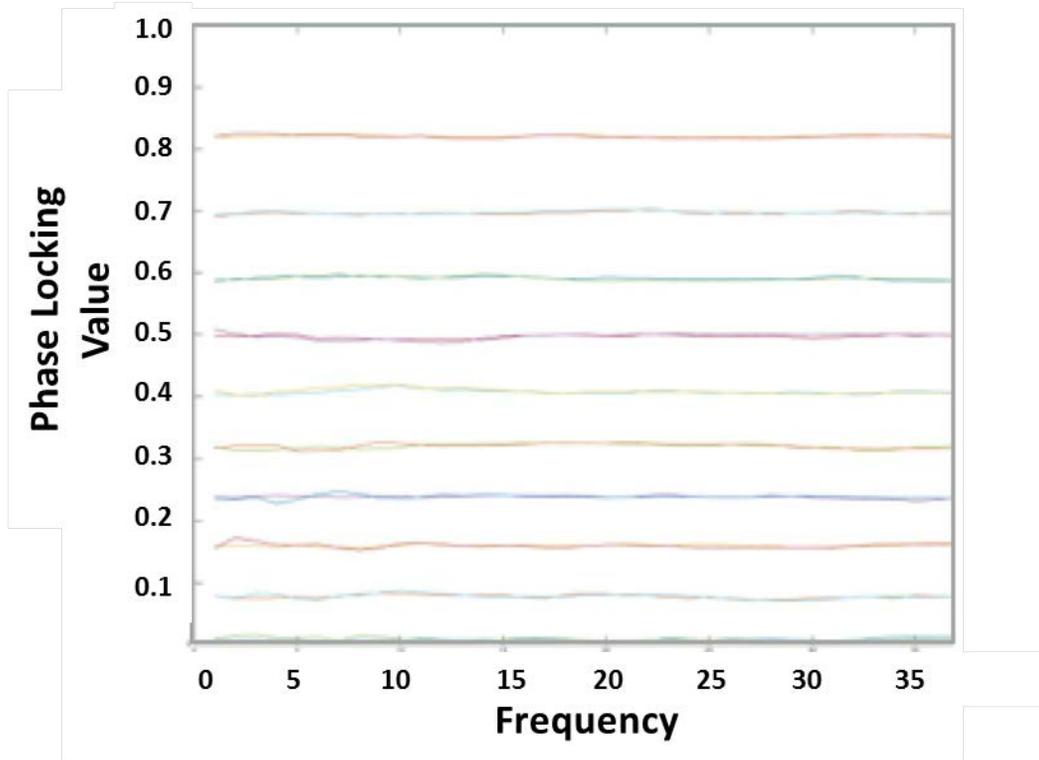
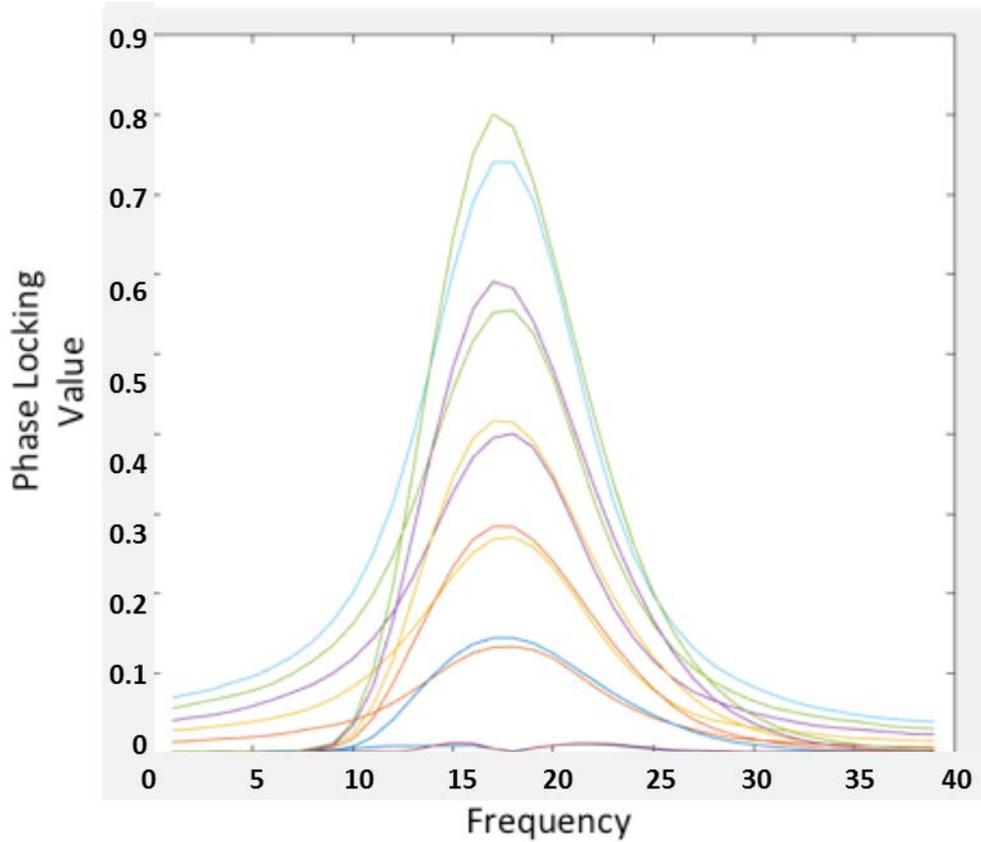


Figure 2-5 PLV values using Hilbert and Morlet methods in Non-Narrow Band Model

PLV values are displayed using Hilbert and Morlet time-frequency methods using a narrow-band model of correlated signals. Higher correlated signals have higher PLV values while lower correlated signals have lower PLV value. There is a good agreement between the Hilbert and Morlet methods across the spectrum. Above, the PLV calculations across frequencies are shown for Hilbert and Morlet wavelet methods. Decreasing correlation in the simulated signals reflect decreasing PLV values across frequencies.

In the above Figure 2-5, we plot of the PLV as a function of frequency for the non-narrow band model using both the Hilbert transform and the Morlet wavelet. There is good agreement between the two methodologies of the PLV value. The signals at the bottom indicate lower correlation while the signals at the top come from high correlation signals. The signals are flat across frequency because there is no frequency-specific correlation.



**Figure 2-6 PLV Values between Hilbert and Morlet in Narrow-Band Model**

Narrow-band model in which highly correlated components of underlying signals only appear at a specific band. Again, Hilbert and Morlet methods have good agreement but now they resolve only a specific window of activity in PLV. Above, the PLV calculations across frequencies are shown for Hilbert and Morlet wavelet methods. Decreasing correlation in the simulated signals reflect decreasing PLV values across frequencies. Figure 2-6 shows the narrow-band model PLV results across frequencies. Here we see that the application of the band-pass filter to the correlated signals produces a narrow window in which the signals have high PLV. Again, we note a good agreement between Hilbert transform and Morlet wavelet methodologies for computing the instantaneous phase at a given correlated signal level. In this

example we note that the PLV value for the highly correlated signal, correlation coefficient of 1, is no longer perfectly phase locked ( $PLV=1$ ) at the center frequency. However, because of our additive noise model, the original signals are no longer perfectly correlated either after application of band-pass filter and the noise. This result demonstrates the ability of PLV to resolve frequency-specific relationships between the signals, whereas the correlation measure across the entire signal is less specific.

Due to the instantaneous phase not being uniquely defined for a given signal, there are multiple methodologies that could be employed for this analysis. However, this section showed the agreement between two common methods when applying PLV. It is important to understand that different choices for the Morlet parameter and different windowing functions (in the above case, the Butterworth filter) will change the instantaneous phase that is obtained.

## 2.5 PHASE LOCKING VALUE STATISTICS

The phase locking value can be calculated mathematically given two signals each with a known distribution of phases. Two sinusoidal time series inputs were used with fixed and equal frequency components and a given distribution of phases. Both signals are sinusoids and have a common frequency,  $\omega$ . The signals are represented as follows:

$$\begin{aligned} s_1(t) &= \sin(\omega t + \phi_1) \\ s_2(t) &= \sin(\omega t + \phi_2) \end{aligned} \quad (2-20)$$

The difference in the phase's random variables are calculated as follows:

$$\phi_d = \phi_2 - \phi_1 \quad (2-21)$$

where  $\phi_d$ , is a random variable of the difference between the two phase distribution random variables  $\phi_1$  and  $\phi_2$ . To calculate the difference between two random variables,  $X$  and  $Y$ , we use  $Z = X - Y$  where  $Z$  is another random variable. The joint distribution of  $X$  and  $Y$  as  $p_{XY}$  is analyzed at each pair of  $i$  and  $j$  where  $i + j = k$ :

$$f_Z(k) = \sum_{\{i,j \mid i+j=k\}} f_{XY}(i,j) \quad (2-22)$$

where  $f_Z$  is the probability distribution function of  $Z$ , and we make the following substitution  $j = k - i$  and obtain the following equation for the difference probability distribution:

$$f_Z(k) = \sum_{i=-\infty}^{\infty} f_{XY}(i, k - i) \quad (2-23)$$

The random variables  $\phi_1$  and  $\phi_2$  are assumed statistically independent. When  $X$  and  $Y$  are treated as statistically independent (i.e.  $p_{XY} = p_X p_Y$ ) the following result is obtained:

$$f_Z(k) = \sum_{i=-\infty}^{\infty} f_X(i) f_Y(k - i) \quad (2-24)$$

Also noting that this is the equation for convolution of two independent functions:

$$f_Z = f_X * f_Y \quad (2-25)$$

where the  $*$  operator denotes the convolution operator. Using this result on our above formulation for the phase differences we obtain the following equation for the difference between the two distributions of phases in signals  $s_1$  and  $s_2$  given the two distributions are statistically independent:

$$f_d(\phi) = \phi_2 * \phi_1 \quad (2-26)$$

To calculate the expected value of the PLV between our two example signals,  $s_1$  and  $s_2$ , the continuous time version of PLV is used:

$$P(\phi_1, \phi_2; T) = \frac{1}{T} \left| \int_0^T e^{i(\phi_1(t) - \phi_2(t))} dt \right| \quad (2-27)$$

for a sampling of time,  $T$ . The following substitution is made:

$$\phi_d(t) = \phi_2(t) - \phi_1(t) \quad (2-28)$$

obtaining the following (with the parameterization by  $T$  dropped to improve the notation):

$$P(\phi_d) = \frac{1}{T} \left| \int_0^T e^{i\phi_d(t)} dt \right| \quad (2-29)$$

The aforementioned can be represented using Euler's formula as:

$$P(\phi_d) = \frac{1}{T} \left| \int_0^T \cos(\phi_d(t)) + i \sin(\phi_d(t)) dt \right| \quad (2-30)$$

Next, the expected value of the analytic simulations must be calculated using the following equation for expected value:

$$E[X] = \int_{-\infty}^{\infty} x f_X(x) dx \quad (2-31)$$

where  $f_X(x)$  is the probability distribution function of the random variable  $X$ . The PLV equation can be refactored as the absolute value of the expected value of the difference between the phase angles. Substituting into the above equation we obtain:

$$|E[e^{i\Phi_d}]| = \left| \int_{-\infty}^{\infty} e^{i\phi} f_{\Phi_d}(\phi) d\phi \right| \quad (2-32)$$

where  $f_{\Phi_d}(\phi)$  is the probability distribution function of the random variable  $\Phi_d$  that represents the difference in the two sinusoidal phases and  $\phi$  is the phase difference angle. Substituting the Euler representation of the PLV equation into the above the following equation is obtained:

$$|E[e^{i\Phi_d}]| = \left| \int_{-\infty}^{\infty} (\cos(\phi) + i \sin(\phi)) f_{\Phi_d}(\phi) d\phi \right| \quad (2-33)$$

This analysis is restricted to probability density functions of the phase differences that are symmetric (i.e. an even function such that  $f(x) = f(-x)$  for all  $x$ ) from which the following result is obtained:

$$\begin{aligned}
 |E[e^{i\Phi_d}]| &= \left| \int_{-\infty}^{\infty} (\cos(\phi) + i \sin(\phi)) f_{\Phi_d}(\phi) d\phi \right| \\
 &= \left| \int_{-\infty}^{\infty} \cos(\phi) f_{\Phi_d}(\phi) d\phi + i \int_{-\infty}^{\infty} \sin(\phi) f_{\Phi_d}(\phi) d\phi \right| \\
 &= \left| \int_{-\infty}^{\infty} \cos(\phi) f_{\Phi_d}(\phi) d\phi + 0 \right| \\
 &= 2 \left| \int_0^{\pi} \cos(\phi) f_{\Phi_d}(\phi) d\phi \right| \\
 &= |E[\cos(\phi)]| \tag{2-34}
 \end{aligned}$$

The second term on the third line is zero because of symmetry requirements. Furthermore, for wrapped phases the integral only needs to be computed between  $[0, \pi]$ . The last result shows that under assumptions of symmetry of the phase differences, PLV is related to the expected value of the cosine of the phase differences between the vectors.

### 2.5.1 Calculation of PLV with Known Phase Distributions

In order to better understand the properties of PLV on signals, the PLV was calculated on a variety of analytic signals with known phases. Distributions of phases for two fixed sinusoids are first examined. Then, this result can be generalized to arbitrarily sized uniform distributions. Finally, uniform distributions of phase differences can be examined which approach PLV of 0 or 1 when the distribution goes from completely random around the unit circle to non-random, respectively.

### 2.5.2 Uniform Distribution of Phase Differences

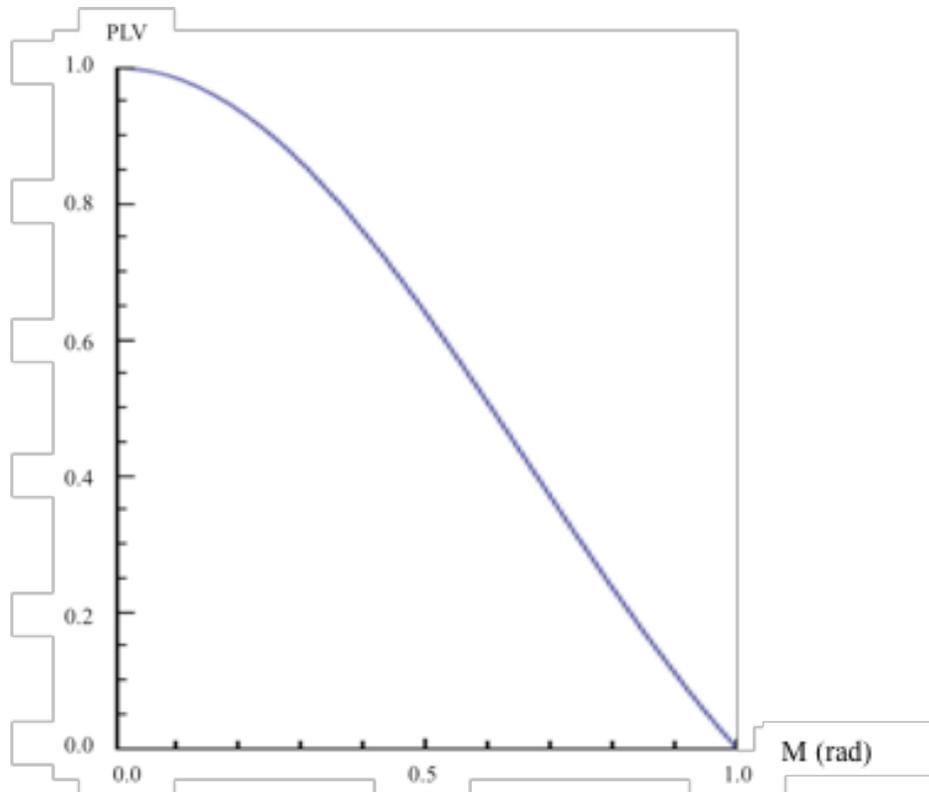
In this example, one sinusoid is a fixed sinusoid at a fixed frequency with no random phase offsets (constant and zero), and a second signal that has uniform phase distribution given by:

$$f_{\phi_d} \sim U(-\pi M, \pi M) \tag{2-35}$$

where  $0 \leq M \leq 1$  and the difference distribution is equal to the uniform distribution. The expected PLV is calculated via:

$$\begin{aligned}
 |E[e^{i\Phi_d}]| &= 2 \left| \int_0^\pi f_{\Phi_d}(\phi) \cos \phi \, d\phi \right| \\
 &= 2 \left| \int_0^{\pi M} \frac{1}{\pi M} \cos \phi \, d\phi \right| \\
 &= \left| \frac{\sin(\pi M)}{\pi M} \right| \\
 &= |\text{sinc}(M)| \tag{2-36}
 \end{aligned}$$

The below Figure 2-7 plots values of M against expected phase locking values.



**Figure 2-7 PLV for expanding uniform distributions**

We can see from the above Figure 2-7, that as the distribution of phase differences becomes completely uniform at  $M = 1$ , the PLV drops to zero. In the opposite direction, as  $M$  approaches a smaller range nearer to zero, the PLV approaches one. This is the desired effect with very narrow distributions of phase differences resulting in a higher PLV.

### 2.5.3 Phase Difference of Two Uniform Distributions

Two sinusoids, each with uniformly distributed phases, are explored next as a representative example of simple sinusoids. The phase distributions of the two signals are given as follows:

$$\begin{aligned}\phi_1 &\sim U(-C, C) \\ \phi_2 &\sim U(-C, C)\end{aligned}\tag{2-37}$$

where  $C$  is a scalar representing the width of the uniform distribution. Using our convolution model for the differences between phases, we obtain a triangle function between the two uniform distributions. The probability density function (PDF) of the difference is:

$$f_{\phi_d}(\phi) = \begin{cases} \frac{\phi}{4C^2} & , \phi \geq 0 \\ -\frac{\phi}{4C^2} & , \phi < 0 \end{cases}\tag{2-38}$$

where  $\phi$  is the phase difference angle (the parameterization of the PDF). The original distributions and the corresponding difference distribution are shown below for generic parameters of the uniform distribution (Figure 2-8).

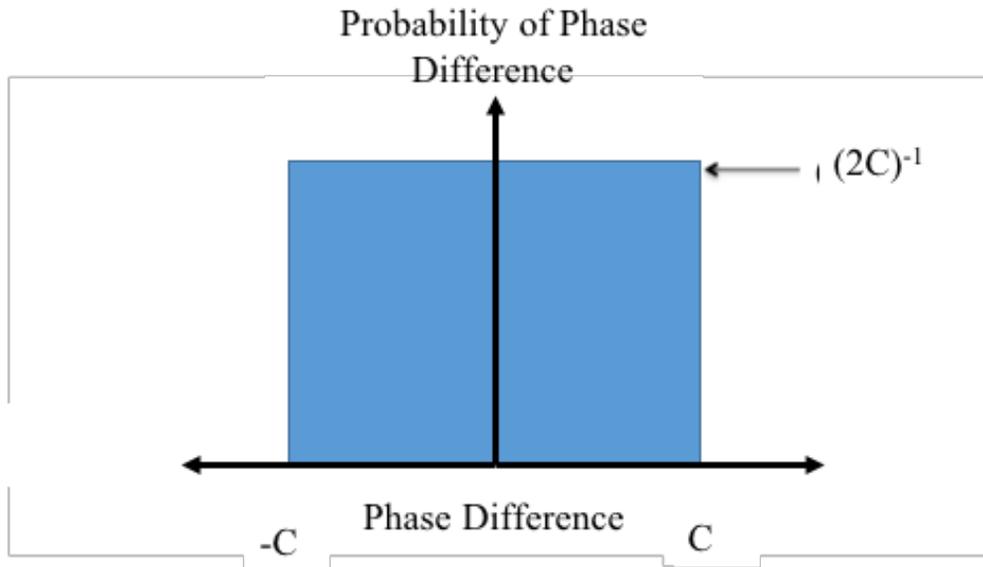
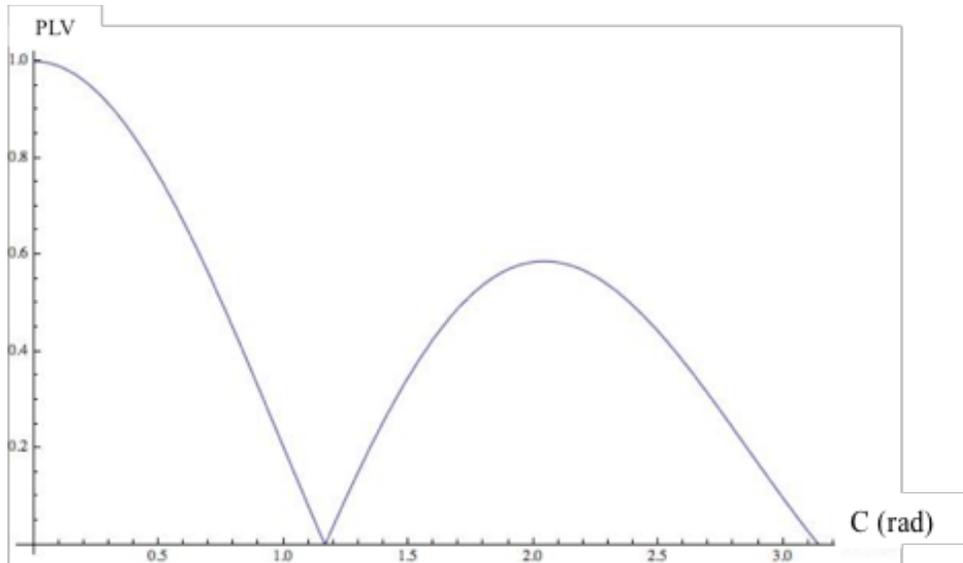


Figure 2-8 A generic boxcar function

Applying this to obtain the expected value of PLV:

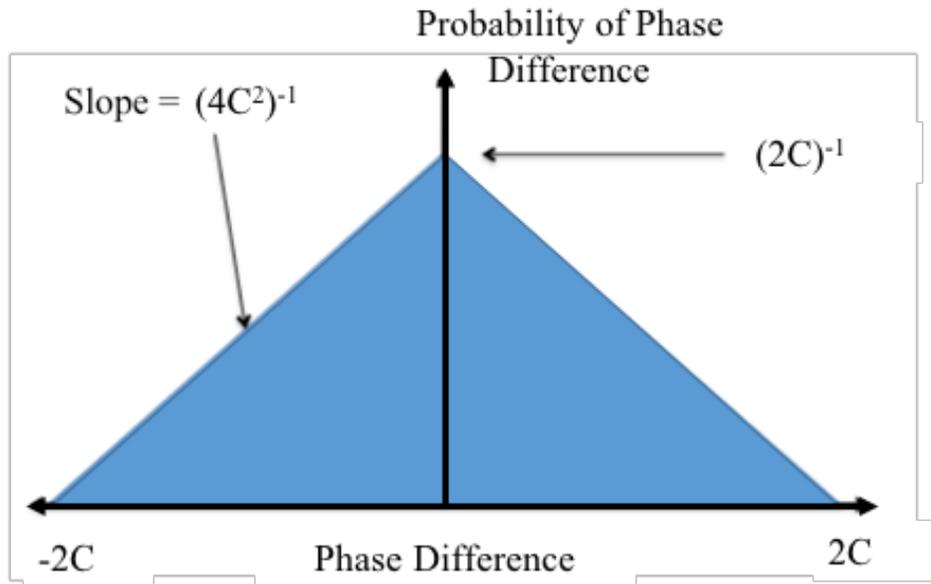
$$\begin{aligned}
 E[P] &= 2 \left| \int_0^{2C} -\frac{\phi}{4C^2} \cos \phi \, d\phi \right| \\
 &= \left| \frac{(\sin C - 2C \cos C) \sin C}{C^2} \right|
 \end{aligned}
 \tag{2-39}$$

The above equation is plotted below in Figure 2-9:



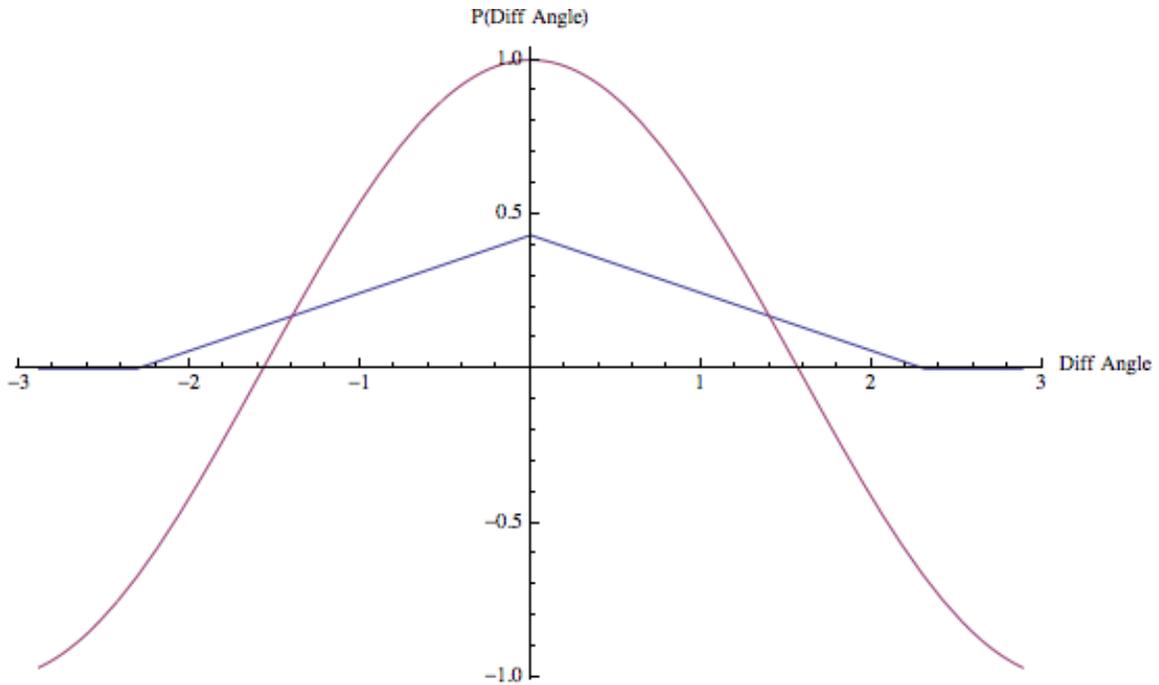
**Figure 2-9 PLV as a function of width parameter of two boxcar phase distributions**

Figure 2-10 shows that as  $C$  approaches zero and the width of the original phase differences becomes very small, the PLV approaches one, indicating a less random phase relationship. As  $C$  becomes larger the convolution of the two boxcar functions begins to spread the phase angles of the vectors over larger amounts of the unit circle and therefore the PLV approaches zero with some oscillations.



**Figure 2-10 Result of convolving two identical independent boxcar functions**

At the first zero point (at roughly  $X \approx 1.165$ ), the triangle distribution is visualized in the following Figure 2-11, where Cosine and width of phase difference random variable are plotted overlaid upon each other to show the interaction of components that lead to sinusoidal behavior of PLV for two boxcar functions.



**Figure 2-11 Probability distribution of phase angles**

The cosine is also plotted in the above Figure 2-11. At  $C \approx 1.165$ , the base of the distribution extends to values greater than  $\pi/2$  while similarly the magnitude of the cosine contribution diminishes and becomes negative. This interplay between the two terms of the integration results in a ringing function of the expected value as a function of  $C$ . This angle wrapping extends to values outside the range  $[-\pi, \pi]$ , the series continues to oscillate with a dampening effect as the phases wrap. As  $C$  tends to infinity, the PLV value tend to zero.

## 2.6 RELATIONSHIP BETWEEN PLV AND CORRELATION

Determining the relationship between PLV and correlation, a commonly employed method of characterizing the relationship between two signals, will allow us to compare methodologies and then interpret the results accordingly. This determination begins with the definition of auto-correlation and cross-correlation:

$$\begin{aligned}R_{xx}(t_1, t_2) &= E \{x(t_1)x^*(t_2)\} \\R_{xy}(t_1, t_2) &= E \{x(t_1)y^*(t_2)\}\end{aligned}\tag{2-40}$$

This definition is disassembled using covariances to obtain the Pearson correlation coefficient equation:

$$\rho_{xy} = \frac{c(x,y)}{\sqrt{c(x,x)c(y,y)}}\tag{2-41}$$

Calculation of covariances is calculated via:

$$\begin{aligned}
C_{xy}(x(t_1)y^*(t_2)) &= E\{(x(t_1) - E\{x(t_1)\})(y^*(t_2) - E\{y^*(t_2)\})\} \\
E\{x(t_1)\} &= E\{y^*(t_1)\} = 0 \\
C_{xy}(t_1, t_2) &= R_{xy}(t_1, t_2) \\
C_{xy}(t) &= C_{xy}(t_1, t_2)|_{t_1=t_2=t}
\end{aligned} \tag{2-42}$$

where we compute the covariance between one of the signals and its conjugate. We further make the assumption that the means are zero. Calculating the correlation and time-averaged correlation from basic sinusoids and their complex analogs the following is obtained after grouping terms and setting the two time offsets to be equal:

$$\begin{aligned}
C_{xy}(t) &= E\{z_x(t)z_y^*(t)\} \\
&= E\{A_x(t)A_y^*(t)e^{j(w_x t + \phi_x(t))}e^{-j(w_y t + \phi_y(t))}\}
\end{aligned} \tag{2-43}$$

Making the similar assumption that the phases between the two signals of interest are equal:

$$\begin{aligned}
w_x &= w_y \\
C_{xy}(t) &= E\{A_x(t)A_y^*(t)\}E\{e^{j(\phi_x(t) - \phi_y(t))}\}
\end{aligned} \tag{2-44}$$

Breaking the expectation operator between the amplitude and phase components reveals that this is the correlation of the analytic signals, not of the original time series. From this, the second term is similar to the PLV result, except for the norm that appears in PLV. For slowly varying or constant amplitudes, the first expectation term creates an amplitude dependence which is lacking from PLV. Hence the difference between the methods is that PLV is not sensitive to amplitude differences or relationships between the signals, only to the phase relationships.

Using the time-averaging methodology introduced above, we next obtain:

$$\overline{C_{xy}(t)} = \frac{1}{T} \int_0^T E\{A_x(t)A_y^*(t)\} E\left\{e^{j(\phi_x(t)-\phi_y(t))}\right\} dt \quad (2-45)$$

And under similar assumptions on the symmetry of the phase distribution, we can further relate to PLV via:

$$\overline{C_{xy}(t)} = E\{A_x(t)A_y^*(t)\} \frac{1}{T} \int_0^T E\left\{\cos\left(\phi_x(t) - \phi_y(t)\right)\right\} dt \quad (2-46)$$

where the second term is replaced with the cosine result found above under symmetry of phase differences between the signals. Again, the same situation arises in which the time-averaged expected values could be interchanged and one must again be careful about non-ergodic and non-wide sense stationary signals. This will again affect the interpretation of experimental results.

To further our understanding of correlation and phase locking, the relationship between constant parameters in our sinusoid examples and PLV is investigated. We start with the assumption of constant phase, amplitude, and equal frequencies and begin by calculating:

$$\rho_{xy} = \frac{C(x,y)}{\sqrt{C(x,x)C(y,y)}} = \frac{C(x,y)}{\sqrt{E\{x^2\}E\{y^2\}}} \quad (2-47)$$

where:

$$C(x,y) = \int_0^T x(t)y(t)dt \quad (2-48)$$

Calculating the numerator first:

$$\begin{aligned} C_{xy}(x,y) &= \int_0^T A_x A_y \cos(\omega t + \phi_x) \cos(\omega t + \phi_y) dt \\ &= \frac{A_x A_y}{2} \int_0^T \cos(2\omega t + \phi_x + \phi_y) + \cos(\phi_x + \phi_y) dt \\ &= \frac{A_x A_y}{2} \int_0^T \cos(2\omega t + \phi_x + \phi_y) dt + \frac{A_x A_y}{2} \int_0^T \cos(\phi_x + \phi_y) dt \\ &= \frac{A_x A_y}{2} \int_0^T \cos(2\omega t + \phi_x + \phi_y) dt + \frac{A_x A_y}{2} \int_0^T \cos(\phi_x + \phi_y) dt \\ &= \frac{A_x A_y}{2} \int_0^T \cos(2\omega t + \phi_x + \phi_y) dt + \frac{A_x A_y}{2} \int_0^T \cos(\phi_x + \phi_y) dt \\ &= \frac{A_x A_y}{2} \int_0^T \cos(\phi_x - \phi_y) dt \end{aligned} \quad (2-49)$$

using:

$$\cos(u) \cos(v) = \frac{1}{2} [\cos(u - v) + \cos(u + v)] \quad (2-50)$$

and computing the auto terms:

$$\begin{aligned} E\{x^2\} &= \int_0^T A_x^2 \cos^2(\omega t + \phi_{x_f}) dt \\ &= \int_0^T \frac{A_x^2}{2} (1 + \cos(2\omega t + 2\phi_x)) dt \\ &= \int_0^T \frac{A_x^2}{2} (1 + \cos(2\omega t + 2\phi_x)) dt \xrightarrow{0} \\ &= \frac{TA_x^2}{2} \end{aligned} \quad (2-51)$$

and similarly for the second term:

$$E\{y^2\} = \frac{TA_y^2}{2} \quad (2-52)$$

we then arrive at:

$$\begin{aligned}\rho_{xy} &= \frac{C(x, y)}{\sqrt{E\{x^2\}E\{y^2\}}} \\ &= \frac{\frac{TA_x A_y}{2} \cos(\phi_x - \phi_y)}{\sqrt{\frac{TA_y^2}{2} \frac{TA_x^2}{2}}} \\ &= \cos(\phi_x - \phi_y)\end{aligned}\tag{2-53}$$

Here again the cosine of the difference arises in the solution. For all parameters fixed and constant, the correlation of the two signals will be determined by the phase difference between them. This is a relatively straightforward example, but by allowing the parameters to take on varying but statistically independent values, increased insight into the relationship can be obtained. Similar to above, starting with fixed and equal frequencies between the signals of interest:

$$\begin{aligned}x(t) &= A_x \cos(\omega t + \phi_x) \\ y(t) &= A_y \cos(\omega t + \phi_y)\end{aligned}\tag{2-54}$$

and assuming the amplitude and phases are all statistically independent but are now random variables instead of constants.

As above, the cross covariance is computed:

$$\begin{aligned}
 C_{xy} &= E\{x(t)y(t)\} \\
 &= \int \int \int \int x(t)y(t)f(A_x, A_y, \phi_x, \phi_y)dA_xdA_yd\phi_xd\phi_y \\
 &= \int \int \int \int A_x \cos(\omega t + \phi_x)A_y \cos(\omega t + \phi_y) f(A_x)f(A_y)f(\phi_x)f(\phi_y)dA_xdA_yd\phi_xd\phi_y
 \end{aligned}
 \tag{2-55}$$

Simplifying the amplitude components:

$$E\{A_x A_y\} = \int \int A_x A_y f(A_x) f(A_y) dA_x dA_y \tag{2-56}$$

and using the following relation:

$$\sin(u) \sin(v) = \frac{1}{2} [\cos(u - v) - \cos(u + v)] \tag{2-57}$$

with the substitution:

$$\begin{aligned}
 u &= \omega t + \phi_x \\
 v &= \omega t + \phi_y
 \end{aligned}
 \tag{2-58}$$

the following is obtained:

$$\begin{aligned}
&= \frac{1}{2} E\{A_x A_y\} \left[ \iint \cos(\phi_x - \phi_y) f(\phi_x) f(\phi_y) d\phi_x d\phi_y \right. \\
&\quad \left. - \iint \cos(2wt + \phi_x + \phi_y) f(\phi_x) f(\phi_y) d\phi_x d\phi_y \right]
\end{aligned}
\tag{2-59}$$

The amplitude terms are independent from the phase terms and are moved outside the integrals. The first integral does not contain any explicit dependence upon time while the second term does. From this it is shown there can be a time-varying relationship of correlation based upon the range in which the measurement occurs. This is investigated next by applying time-averaging techniques:

$$\begin{aligned}
\overline{E\{x(t)y(t)\}} &= \frac{1}{T} \int_0^T \frac{1}{2} E\{A_x A_y\} \left[ \iint \cos(\phi_x - \phi_y) f(\phi_x) f(\phi_y) d\phi_x d\phi_y \right. \\
&\quad \left. - \iint \cos(2wt + \phi_x + \phi_y) f(\phi_x) f(\phi_y) d\phi_x d\phi_y \right] \\
&= \\
&\frac{1}{2T} E\{A_x A_y\} \left[ \int_0^T E\{\cos(\phi_x - \phi_y)\} dt \iint \cos(2wt + \phi_x + \phi_y) f(\phi_x) f(\phi_y) d\phi_x d\phi_y dt \right]
\end{aligned}
\tag{2-60}$$

where the first integral again becomes the cosine under similar assumptions of the symmetry of the phase differences. This is simplified via:

$$\int_0^T E\{\cos(\phi_x - \phi_y)\}dt = E\{\cos(\phi_x - \phi_y)\}T \quad (2-61)$$

assuming the sampling period, T, is equal to the periodicity of the original signals ( $2\pi/\omega$ ):

$$\int_0^T \int \int \cos(2\omega t + \phi_z + \phi_y) f(\phi_x) f(\phi_y) d\phi_x d\phi_y dt = 0 \quad (2-62)$$

where the term becomes zero after integrating over the period. Simplifying:

$$\begin{aligned} \overline{E\{x(t)y(t)\}} &= \frac{1}{2} E\{A_x A_y\} \frac{1}{T} E\{\cos(\phi_x - \phi_y)\} T \\ &= \frac{1}{2} E\{A_x A_y\} E\{\cos(\phi_x - \phi_y)\} \end{aligned} \quad (2-63)$$

the self-terms are then computed, starting from the expectation definitions:

$$\begin{aligned}
 \overline{E\{x(t)x(t)\}} &= \int \int x(t)x(t)f(A_x, \phi_x)dA_xd\phi_x \\
 &= \int \int A_x^2 \sin^2(\omega t + \phi_x)f(A_x)f(\phi_x)dA_xd\phi_x \\
 &= E\{A_x^2\} \int \sin^2(\omega t + \phi_x)f(\phi_x)d\phi_x \\
 \overline{E\{x(t)x(t)\}} &= \frac{1}{T} \int_0^T [E\{A_x^2\} \int \sin^2(\omega t + \phi_x)f(\phi_x)d\phi_x] dt \\
 &= \frac{1}{T} E\{A_x^2\} \int_0^T \int \sin^2(\omega t + \phi_x)f(\phi_x)d\phi_x dt
 \end{aligned} \tag{2-64}$$

noting that:

$$\begin{aligned}
 \int_0^T \sin^2(\omega t + \phi_x) dt &= \frac{1}{2} (t - \sin(\omega T + \phi_x) \cos(\omega T + \phi_x)) \\
 &\quad T = \text{period}, \\
 \int_0^T \sin^2(\omega t + \phi_x) dt &= \frac{T}{2}
 \end{aligned} \tag{2-65}$$

and therefore allowing simplification via:

$$\begin{aligned}\overline{E\{x(t)x(t)\}} &= \frac{1}{T} E\{A_x^2\} \frac{T}{2} \int f(\phi_x) d\phi_x \\ &= \frac{1}{T} E\{A_x^2\} \frac{T}{2} \\ &= \frac{1}{2} E\{A_x^2\}\end{aligned}\tag{2-66}$$

and via symmetry:

$$\overline{E\{y(t)y(t)\}} = \frac{1}{2} E\{A_y^2\}\tag{2-67}$$

Beginning again with the definition of the correlation coefficient and plugging in the original results:

$$\begin{aligned}
\rho_{xy} &= \frac{C_{xy}(x, y)}{\sqrt{\sigma_x^2 \sigma_y^2}} = \frac{E(x, y)}{\sqrt{E\{x, x\}E\{y, y\}}} \\
\overline{\rho_{x(t)y(t)}} &= \frac{C_{xy}(x(t), y(t))}{\sqrt{\sigma_{x(t)}^2 \sigma_{y(t)}^2}} \\
&= \frac{E\{x(t), y(t)\}}{\sqrt{E\{x(t), x(t)\}E\{y(t), y(t)\}}} \\
&= \frac{\frac{1}{2}E\{A_x A_y\}E\{\cos(\phi_x - \phi_y)\}}{\sqrt{\frac{1}{2}E\{A_y^2\}E\{A_x^2\}}} \\
\overline{\rho_{x(t)y(t)}} &= \frac{E\{A_x A_y\}E\{\cos(\phi_x - \phi_y)\}}{\sqrt{E\{A_y^2\}E\{A_x^2\}}} \tag{2-68}
\end{aligned}$$

By fixing the amplitude and phase components as in the previous example, the result reduces to the previous answer. Similarly, the leading numerator term and the denominator are amplitude-only components while the cosine term exclusively represents the phase discrepancies. So again, correlation is amplitude and phase dependent unlike phase locking, which is only phase dependent.

## 2.7 SUMMARY

Time-frequency analysis is limited in its temporal and spectral resolution. In time-frequency analysis, one cannot simultaneously localize both a function and the function's Fourier transform. This limitation has the practical consequence that time resolution must be traded for frequency resolution and vice versa. As such, nearby points in the original temporal sequence will have influences of adjacent temporal points secondary to poor resolution (and similarly in the frequency domain). For phase locking analysis, this implies that adjacent frequency values for the wavelet transform will have some overlapping information that is averaged when calculating the mean phase difference over time.

Phase locking is symmetric, that is the PLV between two time series signals,  $s_1$  and  $s_2$ , is equivalent to the PLV between  $s_2$  and  $s_1$  owing to the formulation of phase locking as the magnitude of phase differences. That is, the vector differences will occur 180 degrees out of phase in the reversed input situation and therefore in the other half of the plane, but the magnitude of the variability will be equivalent.

Perfect sinusoids with constant phase and amplitude over time will be perfectly phase locked across all frequencies. The continuous wavelet transform applied at shifting positions at a given scale will result in a new sinusoidal distribution with fixed offsets at every time point between the two series. As a result, the average phase discrepancy will always be the same in a time-point-by-time-point manner resulting in zero variance and therefore a phase locking value of one.

### 2.7.1 Time-Frequency Resolution: Implications On Phase Locking

Finally, the wavelet's resolution of known signals at various time-frequency parameters of the wavelet was investigated. This was first completed through investigation of the effects of changing wavelet frequency resolution parameter ( $f_0/\sigma_f$ ) on white noise signals. 10,000 pairs of white noise time series signals were generated (5 minutes at 100 Hz sampling rate). Unlike the analytic results from above, this was designed to investigate the numerical results from our analysis methods. This analysis provides insight into how the time-windows used in our time-frequency analysis affect the PLV value.

For each pair, the phase locking value was calculated at integer frequencies between 5 and 40 Hz. For each phase locking value the instantaneous phase values were calculated by utilizing a range of nine integer values ranging from 7 to 47 for the Morlet wavelet parameter.

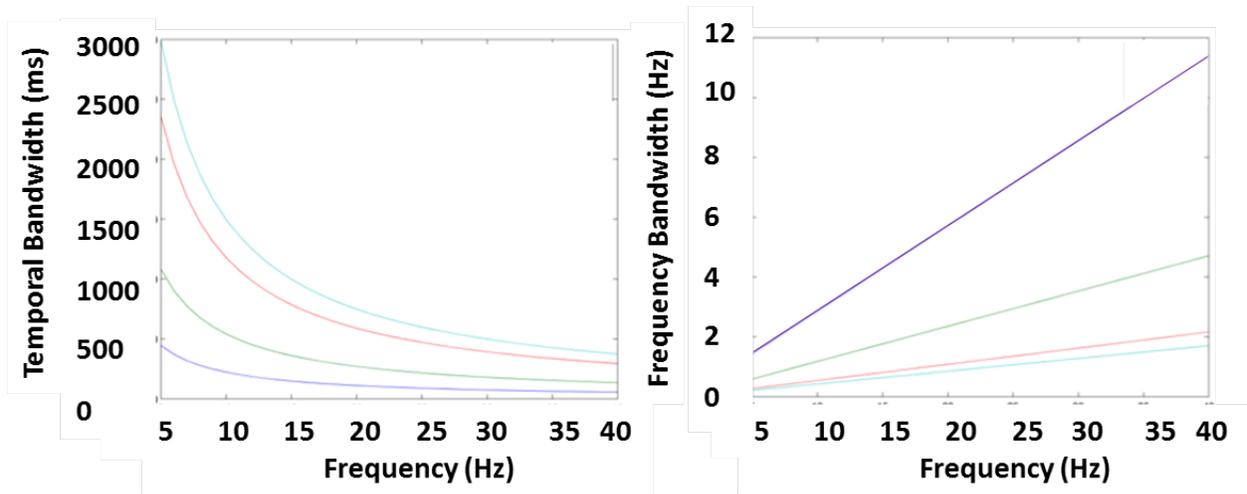


Figure 2-12 Results of wavelet parameters on temporal and spectral bandwidths

Wavelets of size 7 (purple), size 17 (green), size 37 (red), and size 47 (turquoise).

The temporal and spectral bandwidths for different wavelet parameters are shown in Figure 2-12. Results are shown in Figure 2-13 for white noise simulations. Increasing the Morlet parameter increases the phase locking values with a greater effect at lower frequencies. For small width values, the phase locking value was roughly uniform over the frequency ranges shown. At the smallest width, 7, the phase locking value over the range of frequencies shown has a mean value of 0.153 and a standard deviation of 0.04. Note that for increasing frequency widths, the mean phase locking value increases.

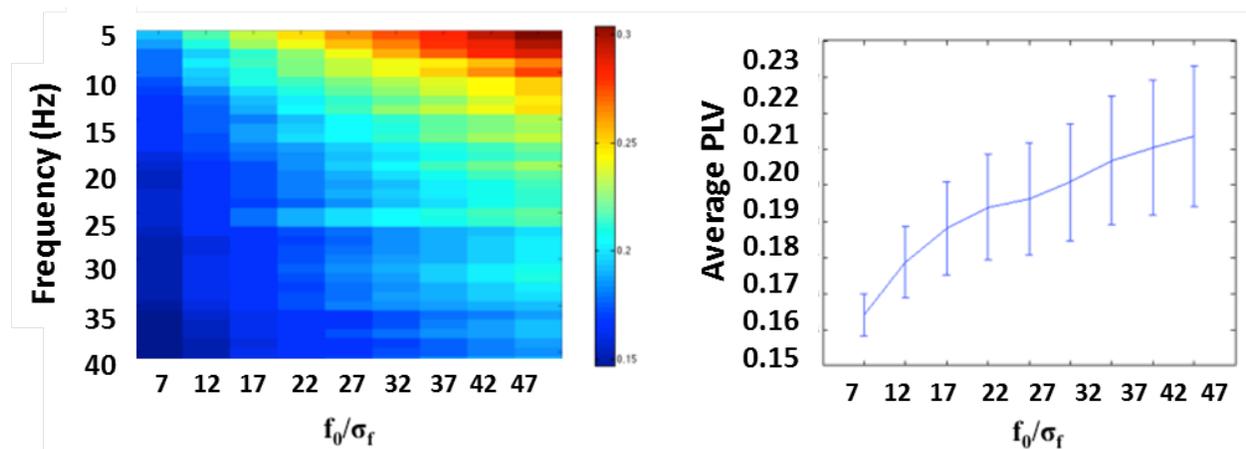


Figure 2-13 Average phase locking value as a function of frequency and wavelet parameter

At lower frequencies, the wavelet has a smoothing effect upon the instantaneous phase results and this becomes more pronounced as the wavelet parameter increases. This smoothing biases the PLV away from zero. A pattern of decreasing phase locking values results from a decrease in the smoothing effects of the wavelet.

Next the result of different wavelet frequency bandwidths was analyzed to determine the effects on the ability to recover phase locked signals. 10,000 pairs of time series signals were simulated. Both signals were generated as a perfect sinusoid with a frequency,  $f_{\text{real}}$  and a random phase offset that was generated by calculating a uniform white noise distribution over the span  $[-\pi/2, \pi/2]$ . This gives rise to two signals that are phase locked. Two frequencies of the underlying signals were used in the analysis, 10 Hz and 25 Hz.

The results of this simulation are shown in Figure 2-14. The top row shows the results for the 10 Hz frequency simulations and the bottom row shows results for the 25 Hz simulations. The wavelet center frequencies, integer values between 5 Hz and 40 Hz inclusive, and wavelet frequency width parameters, 7 to 47, were used as above. The left image in Figure 2-14 shows a representative image of the result of changing frequency bandwidth as a function of the center frequency of the wavelet. The right column images in Figure 2-14 show the averaged results of the 10,000 simulations at each frequency.

In the left column images of Figure 2-14, there is a loss of resolution of the original signals for increasing frequencies that narrows with increasing wavelet parameter. In the right column, averaging over many simulations lowers the resolution of the phase locked signals with increasing wavelet width. As such, lower frequencies are more susceptible to the reduced resolution at higher wavelet parameters as a result of higher baseline phase locking. Ability to resolve a single sinusoid with random phase in a given range. The left column shows a single realization of the 10,000 trials while the right column shows the average result of 10,000 trials.

## Time-Frequency Resolution Dependence Upon Wavelet Frequency Bandwidth

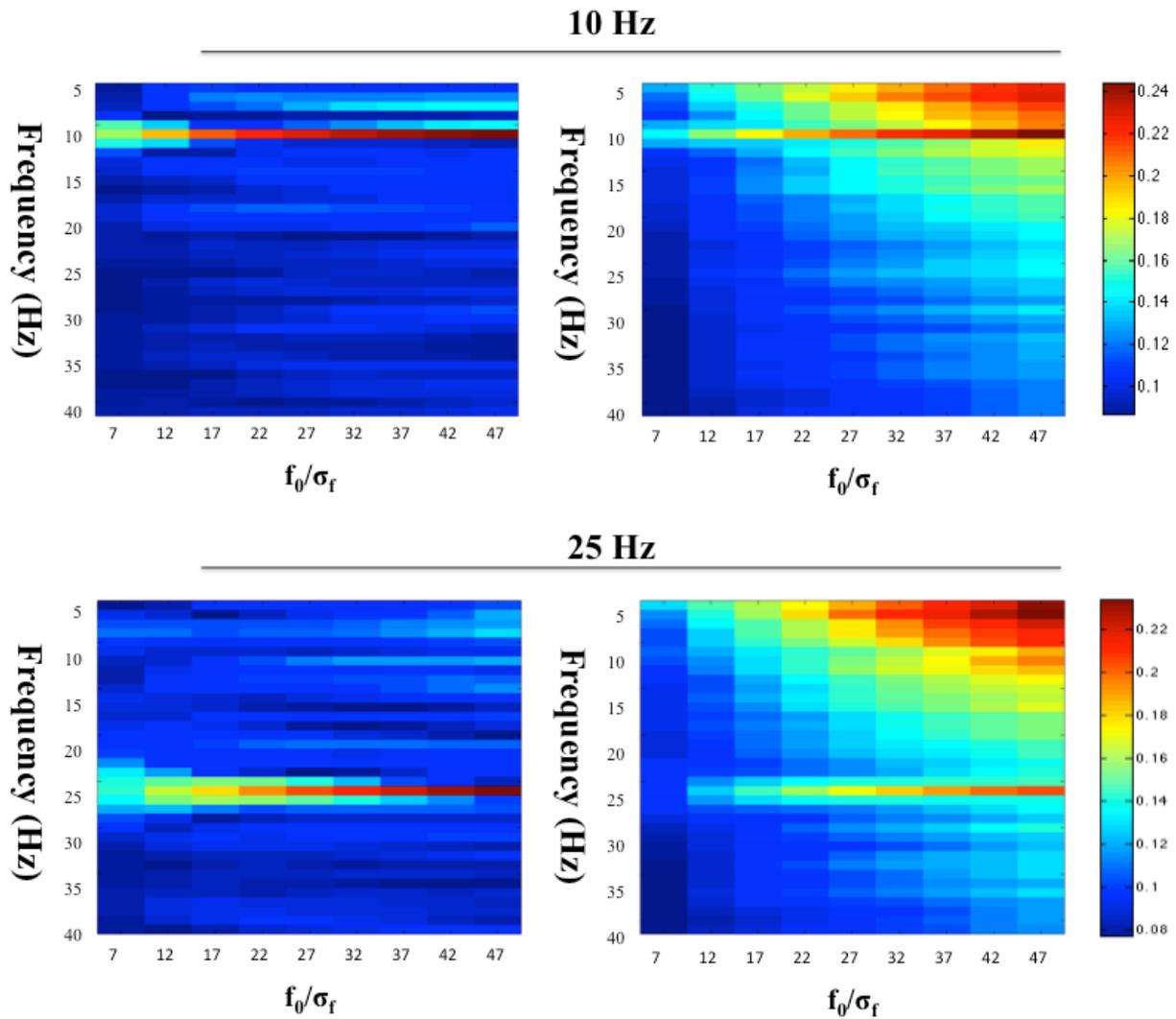


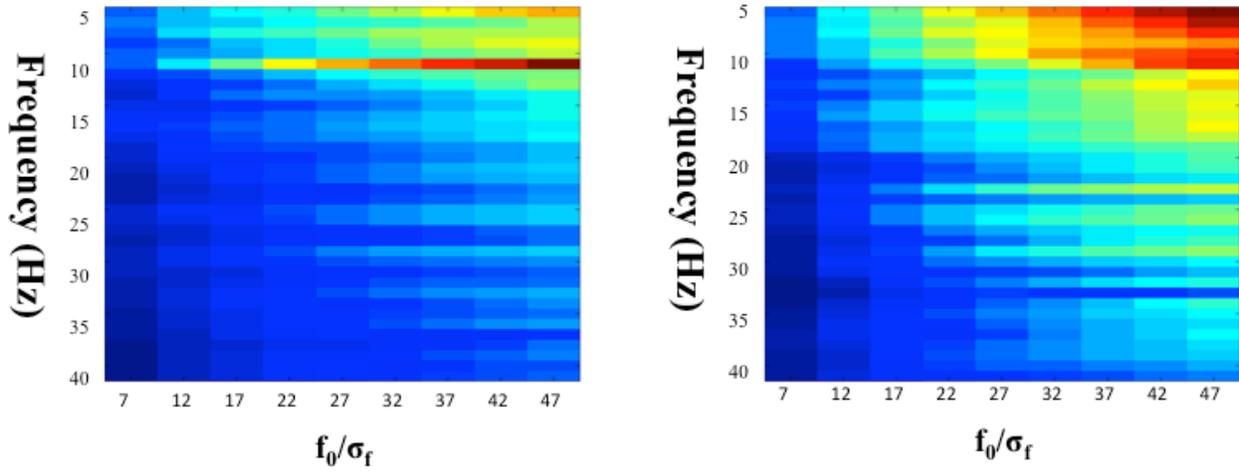
Figure 2-14 Time-Frequency resolution dependence upon wavelet frequency bandwidth

The above numerical results can be compared with the analytic solution produced earlier. In that earlier solution, two sinusoids with uniform phase distributions over the range  $[-\pi/2, \pi/2]$ . In that analytic result, the expected phase locking value is  $4/\pi^2 \approx 0.41$  while the numerical results are maximum at a  $\sim 0.24$ . There are many contributing factors that are likely. Primarily, in the analytic solutions we are assuming perfect reconstruction of instantaneous phase as well as perfect sampling, neither of which are present in the numerical studies.

Next, the addition of white noise was simulated onto the phase locked sinusoids. Two levels of noise were added corresponding to an SNR of 2 and 10. The underlying phase locked signals were generated as above using 10,000 simulation pairs. Results of this simulation are shown in Figure 2-15. The addition of noise created larger phase locking regions corresponding to lower frequencies that became more pronounced for higher wavelet parameters. Ability to resolve a single sinusoid with random phase in a given range and the addition of random noise. The left column shows a single realization of the 10,000 trials while the right column shows the average result of 10,000 trials.

## Time-Frequency Resolution Dependence Upon Wavelet Frequency Bandwidth With Noise

10 Hz



25 Hz

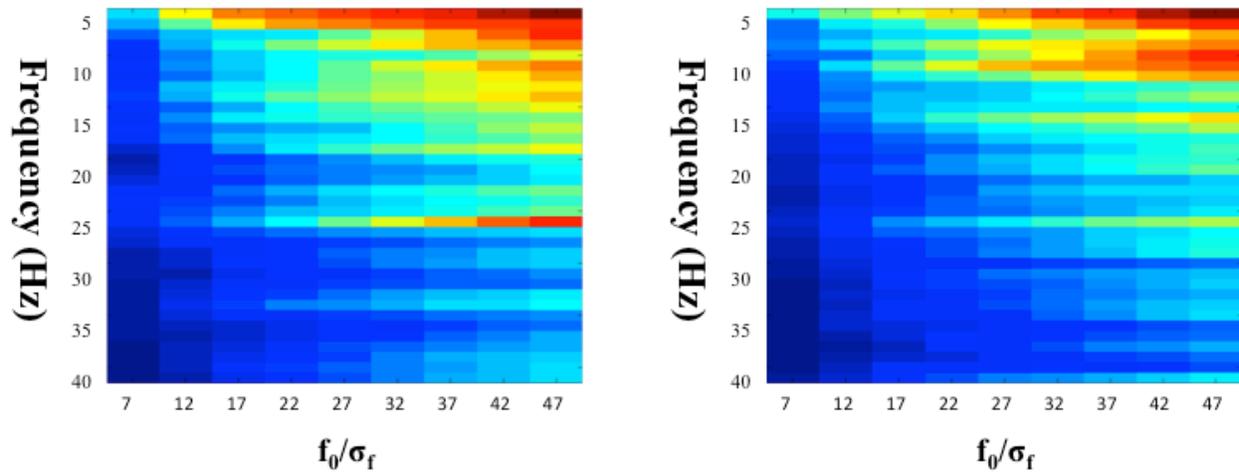


Figure 2-15 Resolution dependence upon wavelet frequency bandwidth with noise

At higher wavelet parameters, there were a large number of false positive phase locking values. However, the addition of noise tended to degrade the resolution of all frequencies that degraded quickly for smaller wavelet parameters. This is a direct result of the tradeoff between temporal and spectral resolution of wavelets. At higher frequencies, the wavelet's frequency bandwidth increases. However, for higher wavelet parameters, this tradeoff is less rapid. This is seen by the narrowing of the response as a function of higher wavelet parameters. As the wavelet parameter gets larger, the bandwidth at a given center frequency is smaller which results in less blurring. Conversely, for higher wavelet parameters the lower frequencies tend to have a higher phase locking value. This is a result of the larger temporal bandwidth with higher wavelet parameters. From Figure 2-15, this effect is less pronounced at higher frequencies as the temporal duration converges for the selected wavelet parameters. From this analysis, the use of wavelet parameter of 7 allows for the analysis across the range of physiological frequencies: 5-30 Hz.

## 2.8 CONCLUSION

Phase locking implicitly depends on the notion of being able to estimate an instantaneous phase as a function of time. This is important in the study of neurophysiological systems in order to have the ability to analyze the temporal and spectral components of a signal. As mentioned above, phase locking is one of a variety of methods for estimating the instantaneous phase of a signal. There exist a wide variety of wavelets that can be used in lieu of the Morlet wavelet. They each have their own properties for understanding temporal and spectral properties of signal. The Morlet wavelet is commonly employed in neuroimaging literature allowing direct comparison of results. In addition, the Morlet is especially easy to work with because of the ability to re-parameterize it to make center frequencies explicit.

Our results show numerical and analytical results of the properties of PLV using both Morlet wavelets and Hilbert transforms to obtain instantaneous phases. The choice and configuration of the selected methodologies will have an impact on any experimental results and it is important to understand the tradeoffs and considerations in each of these choices.

### 3.0 PHASE LOCKING GRAPHS

In this chapter, we created a framework for the analysis of functional connectivity estimates in large-scale neural recordings. Phase locking was calculated between large numbers of neural populations yielding phase locking graphs. Methods to test for statistically significant networks were established which considered the implication of multiple statistical inferences in assessing statistically significant networks.

#### 3.1 INTRODUCTION

The brain consists of roughly  $10^{11}$  neurons and approximately  $10^{14}$  synaptic connections between those neurons [42]. This complex network of anatomical connections, as stated previously, is currently impossible to directly observe. As such, an understanding of the statistical relationship between multiple neural populations can provide insight into the underlying architecture of the brain.

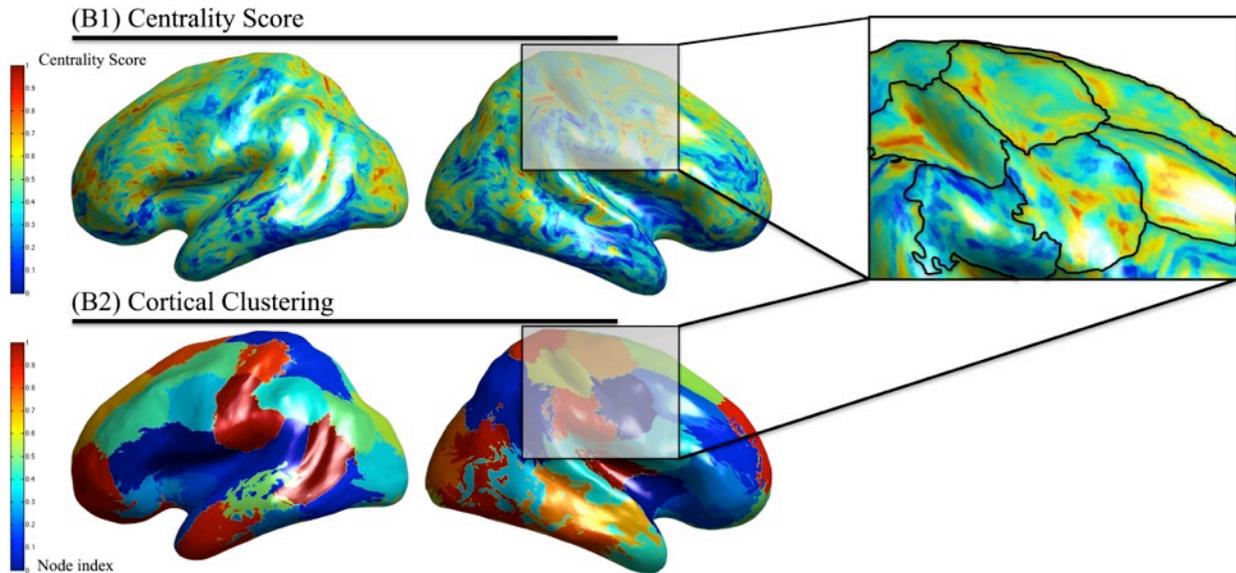
Functional connectivity allows researchers to investigate the connection dynamics associated with neural populations. However, the investigation of large-scale neural dynamics necessitates that the analysis be conducted at multiple sites simultaneously [43]. Confined analysis to specific regions of interest (ROI) permit targeted experiments regarding the specific functional connectivity between a small subset of the overall network architecture. This allows

traditional statistical methods to be employed for statistical testing. Furthermore, these ROI investigations allow the integration of previous knowledge as priors of the neurophysiological mechanisms.

Many imaging modalities commonly employed in functional connectivity analysis produce large quantities of data creating an explosion of information regarding neural processes. The simplest data reduction technique is to limit analyses to specific regions under investigation for a given experiment. However, these technical limitations are being overcome by the combination of increasing computational power and advanced software packages for handling the large scale of neuroimaging data.

The use of multiple simultaneous statistical inferences for the analysis of whole brain functional connectivity reduces the power of those inferences. In some cases, overly conservative methods of correction can eliminate any ability to resolve positive findings resulting in a larger number of false negatives.

However, the investigation of whole brain neural dynamics is an important area of study. Investigations utilizing multiple ROIs and small-scale network dynamics have already made promising discoveries regarding the architecture of healthy brains [43]. In addition, the use of multiple ROIs has yielded an improved understanding of the nature and effects of disease states upon functional connectivity networks [43-47], insights which could lead to new diagnostic tools or an improved understanding of these disease models. Figure 3-1 shows a sample of the results available to the methodology presented here. The data-driven identification of functionally connected regions and their statistical relationship can be understood by employing graph theory methodologies which are explained in depth below.



**Figure 3-1 Identification of Functionally Defined Regions**

In this chapter we have created a methodological framework for approaching the issues associated with large-scale whole brain phase locking functional connectivity. Methods were developed for efficiently analyzing large-scale functional connectivity networks. Using the mathematical foundations of graph theory these networks can be investigated and quantified. Finally, non-parametric methods were employed to identify statistically significant network connections occurring within these large-scale networks while explicitly incorporating the multiple comparison issue into the analysis.

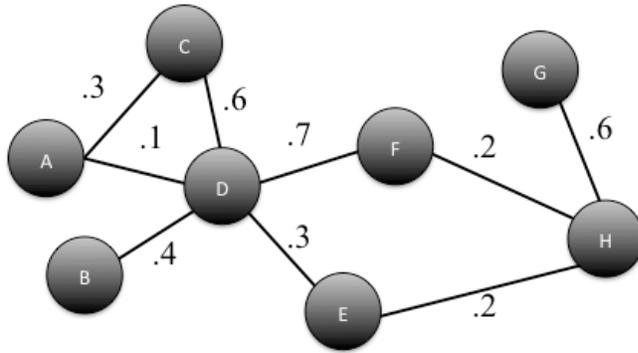
## 3.2 BACKGROUND

Graphs are composed of objects and the relations between those objects. They are typically represented as  $G=[V,E]$  where the graph,  $G$  contains a set of vertex objects,  $V$ , and edges,  $E$ . Each edge represents a relationship between two vertices [48, 49].

Graphs can be characterized broadly into four categories corresponding to two properties of their edges. Edges which relate two vertices,  $V_1$  to  $V_2$  can be either undirected or directed and can also be weighted or unweighted. Undirected edges represent a relation between two vertices  $V_1$  and  $V_2$  in which the order of the relation is symmetric and therefore no order is imposed. Directed edges contain a relation between two edges that is unidirectional. For instance, a directed edge might relate  $V_1$  to  $V_2$  but does not represent any information regarding  $V_2$  to  $V_1$ . Unweighted edges represent a binary relation between two vertices such that the edge exists or does not exist while weighted edges contain an edge value representing the strength of the relation between vertices.

Graphs can be visualized via a diagram in which vertices are represented as nodes and edges are represented as lines or arrows between them. An example is given in Figure 3-2. Graphs can also be represented as an adjacency matrix which is a  $n \times n$  square matrix where  $n$  is the number of vertices and each entry in the matrix  $[i, j]$  is the graph weight (in the case of weighted graphs, for unweighted this is a binary indicator of edge existence) between the  $i^{\text{th}}$  and  $j^{\text{th}}$  vertices. The adjacency matrix diagonal represents self-connections of a vertex with itself.

A



B

	A	B	C	D	E	F	G	H
A			0.3	0.1				
B				0.4				
C	0.3			0.6				
D	0.1	0.4	0.6		0.3	0.7		
E				0.3				0.2
F				0.7				0.2
G								0.6
H					0.2	0.2	0.6	

**Figure 3-2 Graph Representation.**

Figure 3-2 (A) is a graph in which circles are vertices with associated letter labels. Connections between vertices are edges and are represented as lines between vertices. Each edge also has a weight label. In Figure 3-2 (B) we have the adjacency matrix representation of the graph in Figure 3-2 (A). The values of the matrix are the values of the edges between the row and column vertices.

Phase locking graphs represent the functional connectivity data linking neural populations. Graph vertices are derived from neural population recordings. The phase locking value from each vertex to every other vertex are calculated and used as edges with weights between the vertices. Phase locking is symmetric between the two time series being tested; therefore, phase locking graphs are undirected weighted graphs.

In addition, vertex values have additional meta-data. Each vertex has a set of 3D coordinates that indicate that vertex's spatial coordinates. For our purposes, the spatial coordinates are localized specifically to the cortical surface and not to deeper brain structures (see section 5.3.1).

The cortical surface can be inflated into a 2D spherical surface in a 3D space by use of an inflation transform. This transform is covered later in chapter 5. For now, each vertex can have spatial coordinates that are transformable between the anatomical cortical surfaces or via the spherical coordinates.

Each phase locking graph can also be represented as an adjacency matrix. In general, it is impossible to transform a 3D surface into a sorted nearest-neighbor 1D list, as a result the vertex ordering in an adjacency matrix will not have the property that nearby vertices are spatially sorted according to closeness in the cortical surface space. However, the associated meta-data is still retained. This restriction makes the interpretation of adjacency matrices more difficult with respect to direct neuroanatomical structures. Additionally, the adjacency matrix is symmetric about the diagonal due to the symmetric nature of phase locking.

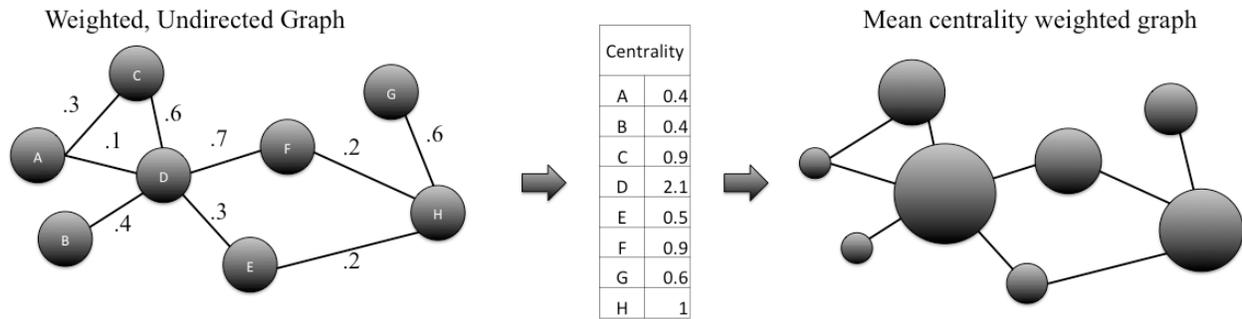
Because of the nature of the Rayleigh distribution of phase locking values, the chance of zero phase locking is relatively small. Therefore, phase locking graphs tend to be complete where a complete graph is one in which each vertex is connected to every other vertex.

## 3.3 METHODS

### 3.3.1 Network Centrality

The centrality of a vertex is a measure of the importance of that vertex within the graph. Many methods exist for calculating the centrality of each vertex in a graph including degree centrality and eigenvector centrality.

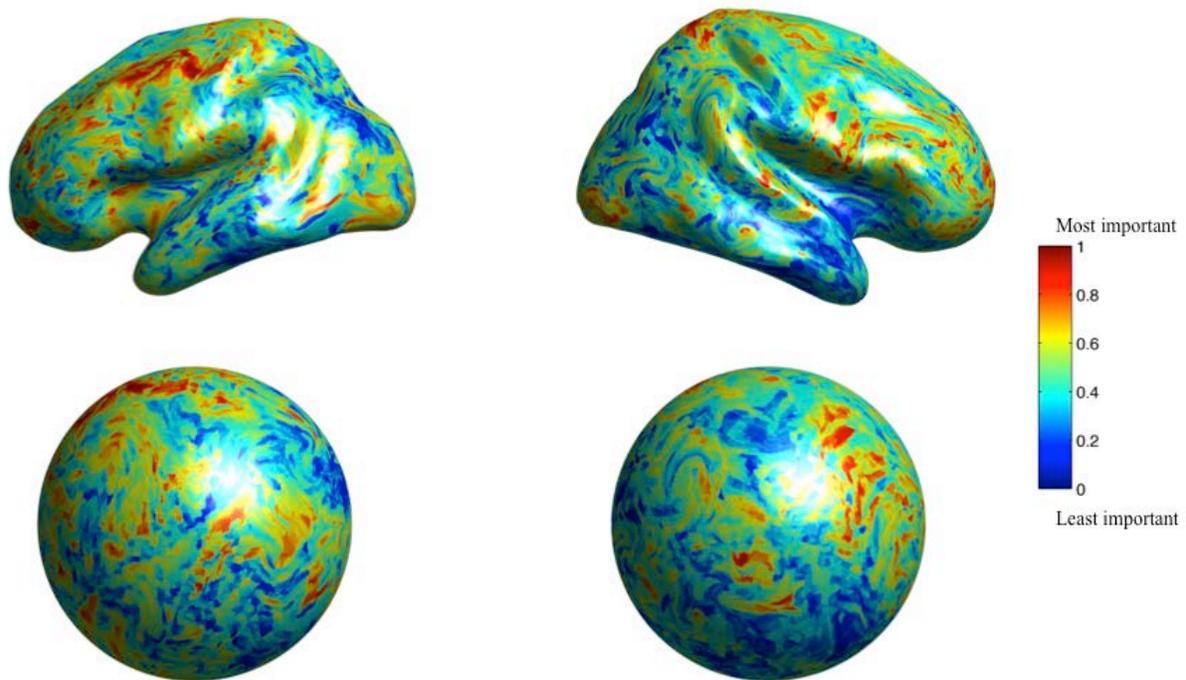
Degree centrality is the simplest measure of node importance. It assigns the number of connected vertices to a given vertex as the centrality score. For phase locking graphs, it is the sum of weighted incident edges from the node of interest. Degree centrality can be calculated for each vertex by summing the weights of every edge incident with that vertex (i.e. summing down the row or across the columns of the adjacency matrix). In addition, dividing each centrality measure by the maximal centrality score can normalize it. In this way, the degree centrality is relative to other vertices.



**Figure 3-3 Graph Centrality.**

On the left of Figure 3-3, is an undirected weighted graph. The center image is a calculation of the degree centrality of each vertex. The degree centrality for a vertex is calculated by summing the weighted edges incident on the given vertex. On the right, is a graph in which vertex size is proportional of the centrality value.

Eigenvector centrality (EVC) is another measure of importance that relies upon the dominant directionality of the two-dimensional adjacency matrix. For a given vertex, it incorporates both the number and importance of the vertices it connects to. Eigenvector centrality is defined as the first eigenvector of the adjacency matrix of the graph.



**Figure 3-4 Eigenvector Centrality Real Phase Locking Graphs.**

In Figure 3-4, the top is the eigenvector centrality of a phase locking graph displayed on an inflated cortical surface. Bottom, is the same values of the eigenvector centrality but on a spherically inflated surface. A reversible transform is possible between the top and bottom cortical surface representations allowing visualization in either space.

Centrality measures are a useful method for data reduction. In a large graph structure, with  $N$  vertices, the number of connections in an undirected graph is  $(N)(N-1)/2$ . However, centrality measures provide a data reduction technique that results in a data vector of size  $N \times 1$ . The display of an  $N \times N$  matrix with the associated meta-data is not possible as it is inherently a 7-dimensional vector ( $x, y, z$  coordinates of both vertices and the phase locking value). The  $N \times 1$  centrality scores, however, can be displayed directly onto the original source space vertices in the 4-dimensional space.

The output of different centrality measures demonstrates the importance of nodes within the graph. It reduces the information contained within the graph such as the exact nature of network distribution but does provide a visualization of the importance of vertices. Graph data reduction techniques are important for visualization of the information as well as further analysis methods.

### **3.3.2 Functionally Defined Seed Regions**

Brain regions under observation in phase locking graphs might include areas like the Brodmann Areas named after the German anatomist Korbinian Brodmann who, based upon cytoarchitectural boundaries between cortical brain regions, identified distinct cortical areas in monkeys, humans and other species [50]. These cytoarchitectural differences referred to differences in cellular composition observed under a microscope.

In contrast to the cytoarchitecturally defined regions of Brodmann Areas, one can also identify functionally defined regions [28]. Identification of these regions requires ascertaining multiple neural populations that exhibit similar properties in a functionally defined area.

To create phase locking defined functional areas, we began by computing the eigenvector centrality of the phase locking graph. The  $N \times 1$  centrality measure was then combined with the meta-data containing the 3D coordinates of each vertex cluster in source space. Clustering of this  $N \times 4$  dimensional data space yielded functionally defined phase locking regions within spatially smooth clusters upon the cortical surface. These clustering algorithms also helped to reduce the overall dimensionality of the graph data.

The four-dimensional data vector for each vertex location consisted of the three-dimensional meta-data coordinates of the vertex  $(X,Y,Z)$  and the centrality score for that vertex as calculated via eigenvector centrality.

We used a k-means clustering algorithm to calculate the clusters in the eigenvector centrality vector space. K-means clustering attempts to find  $k$  independent clusters given an  $n$ -dimensional input data space. In general, k-means finds clusters of similar extent, such that each cluster is similarly shaped in the  $n$ -dimensional input space and also enforces spatially smooth output cluster spaces. K-means generates output spaces that are Veroni cells in that the entire input space belongs to one of the  $k$  clusters.

K-means partitions the input data space into k clusters each possessing a centroid corresponding to a data point that is nearest to the center of mass of each cluster. It can be solved via the following equation:

$$\mathit{arg\ min}_S \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|^2 \quad (3-1)$$

where S is the configuration of the centroids that minimizes the L2 norm of distances between each input value within the i<sup>th</sup> cluster to the centroid of that cluster,  $\mu$ .

An iterative solution was used in which observations are reclassified into their closest centroid. Seed points were initialized by randomly selecting k data vertices from the input set to initialize the k clusters.

The iteration proceeds in two steps. First, each point is assigned to its closest centroid given the distance metric. Then, each cluster's centroid is recalculated given the new clustering and the closest data point to the new centroid is assigned as the centroid for the next iteration. This method continues until convergence is reached in which the centroid assignments are unchanged between iterations or the changes are within a small tolerance.

The above equation utilizes the Euclidean distance, or L<sub>2</sub> norm. In general, k-means can use any distance function of interest when performing the iterative algorithm for minimization of the within cluster distances. As mentioned previously, the vertex meta-data includes both the cortical surface points as well as their spherical transform coordinates.

Difficulties arise when using cortical surfaces and  $L_2$  norm estimates of distance. Namely, that the folding of the cortical mantle into gyri and sulci can create shortest distance vertices which lie across the sulci divides. In this case, the nearest point in an  $L_2$  sense can be disconnected when following a path along the cortical surface. Using spherical coordinates eliminates this issue since the inflation procedure unfolds the cortical surface, therefore nearby points on the spherical surface are close on the cortical surface.

Vector space representations of N-dimensional vectors with unit spherical magnitudes lend themselves to cosine distance metrics since only angle information is important for determining the distance between two points. The cosine distance implicitly assumes nearby locations on the cortical surface are close by only calculating the angle between them. In contrast,  $L_2$  norm distances in N-dimensional spherical coordinates do not account for the contour of the spherical surface and instead bisect the sphere for distance calculations.

The cosine distance can be derived from the Euclidean dot product formula:

$$A \cdot B = \|A\| \|B\| \cos \theta \quad (3-2)$$

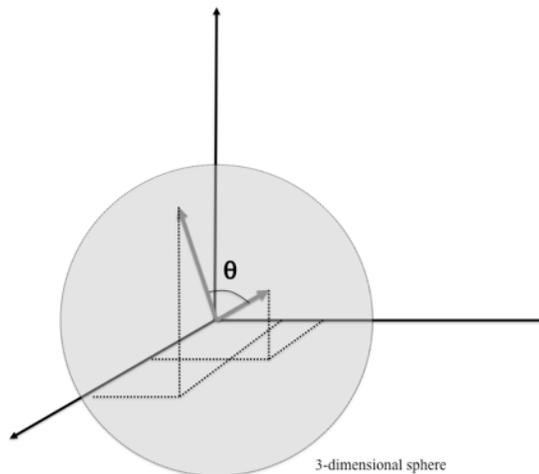
by solving for the cosine term in which case we obtain:

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} \quad (3-3)$$

where A and B are N-dimensional vectors and  $\| \cdot \|$  indicates the Euclidean norm of the vector.

The cosine transform also implicitly accounts for the uniformity of the spherical coordinates, namely that the 3D coordinates can be reduced to a polar representation using only two values.

This is extended to phase locking by incorporating the 4-dimensional input data.



**Figure 3-5 Cosine distance.**

The cosine distance can be calculated in n-dimensional space. Figure 3-5 shows a 3-dimensional demonstration of two vectors (solid grey) incident upon a unit sphere (grey) showing the distance between the two vectors as the angle between them.

Alternatively, a polar coordinate transform could have been explicitly used to transform the Cartesian coordinate sphere into  $(r, \theta, \phi)$ , however this would have produced estimation errors. The transform to polar coordinates would represent a loss of precision in both directions of the transform which would have necessitated either interpolation or estimation techniques. The cortical surface used in practice has ~150,000 vertices therefore estimation errors can lead to poor clusters. Estimation of cosine distance maintains the fixed locations of the vertices without a loss of precision or the need for interpolation. Therefore, it was determined that the cosine distance was the better alternative.

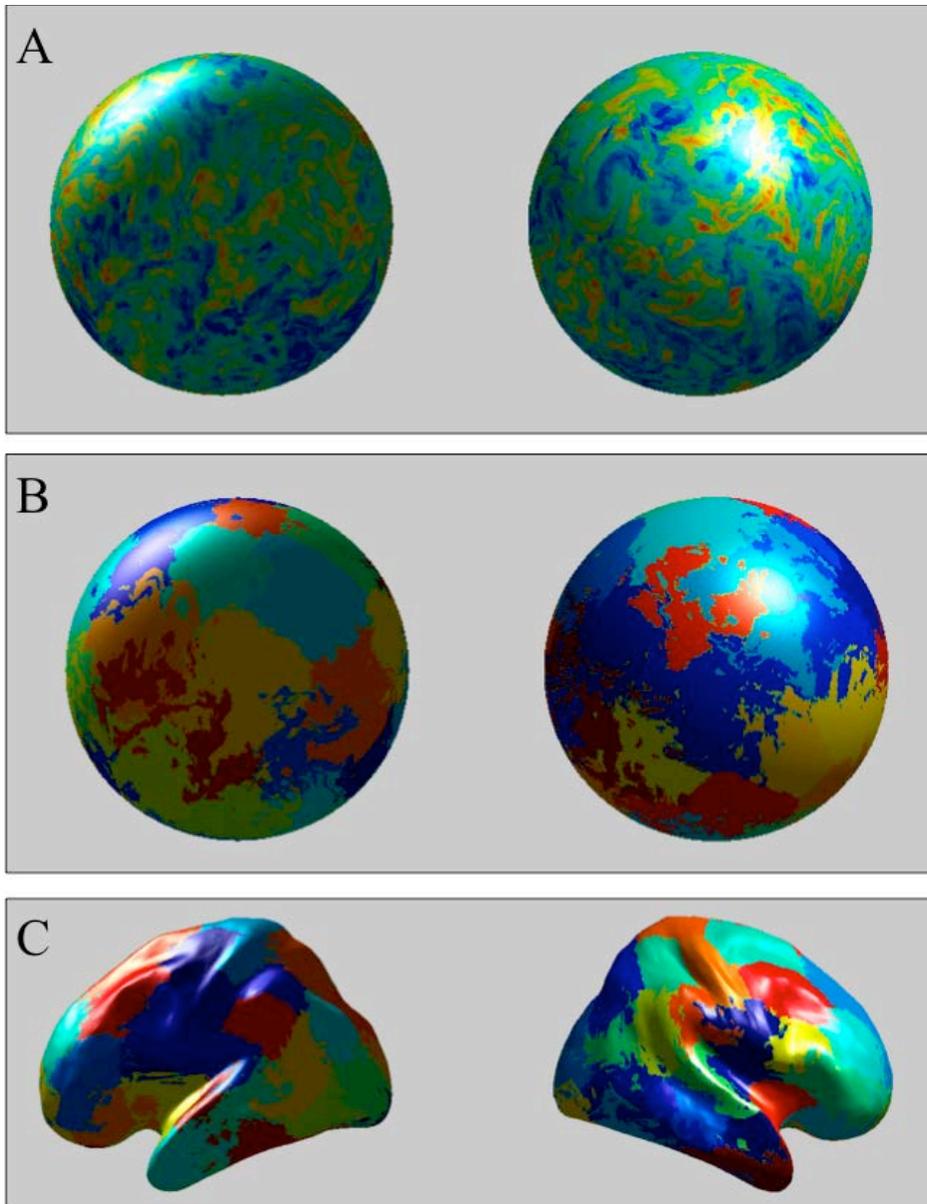


Figure 3-6 Spherical Clustering.

Figure 3-6 (A) shows a display of the eigenvector centrality on spherical cortical surface coordinates. Figure 3-6 (B) is the result of k-means clustering of Figure 3-6 (A) using  $k=40$  for each hemisphere. Colors represent the index of the cluster with blue as the first cluster and red the 40<sup>th</sup> cluster in each hemisphere. In Figure 3-6 (C) the same clustering from Figure 3-6 (B) but on the cortical surface inflated space.

## 3.4 NETWORK ANALYSIS

### 3.4.1 Large Scale Statistical Testing

Traditional statistical methods rely upon a single or small number of simultaneous statistical inferences being made in an experimental setting. As an experiment considers more than a single hypothesis, the chance of a false positive occurring by random chance simultaneously increases. As such, when two or more simultaneous hypotheses are tested, it is necessary to correct for the number of comparisons by modifying the criterion for significance.

For example, in a basic statistical testing paradigm a null hypothesis is asserted that the means of two populations are equal. The test is then conducted to test whether given the observed data of the two populations we should reject the null hypothesis or we cannot reject the null hypothesis. This is done at an *a priori* significance level,  $\alpha$ , from which our test derives a p-value to be compared with the significance level. That p-value is the probability of obtaining our test statistic from the observed data or a value at least as extreme under the assumption that the null hypothesis is true.

For a given significance level and accompanying test statistic, the significance level determines the probability of a type I error occurring for a single hypothesis test. A type I error occurs when we incorrectly reject the null hypothesis, also called a false positive.

When conducting multiple simultaneous comparisons, this significance level can inflate the family wise false positives by random chance. For a given critical value of  $\alpha = .05$ , we have a 5% chance of making a type I error. If we perform a family of 100 such tests, then by chance alone we expect 5 test statistics to falsely reject the null hypothesis.

For independent statistical tests, the family wise error rate (FWER) is defined as the probability of making one or more false discoveries (i.e. one or more type I errors). For independent statistical tests, the probability that one test is significant is (1-probability that none of them are correct) which under the assumption of independent tests is the product of the individual probabilities. Therefore, the FWER is given via:

$$\bar{\alpha} = 1 - (1 - \alpha_{comparison})^n \quad (3-4)$$

where  $\bar{\alpha}$  is the FWER,  $\alpha$  is the significance level of each independent test, and  $n$  is the number of comparisons. For the most conservative approach of multiple comparison correction, we can use:

$$\alpha_{comparison} = \bar{\alpha}/n \quad (3-5)$$

which is the Bonferroni correction. However, this is the most conservative correction and does not take into account the implications of dependent hypothesis tests. A separate correction can be used by solving the above equation for  $\bar{\alpha}$  called the Sidak correction.

$$\alpha_{comparison} = 1 - (1 - \bar{\alpha})^{1/n} \quad (3-6)$$

However, neuroimaging modalities are not independent measurements. In particular, imaging methods that require the use of an inverse model for their solution in source space are highly dependent (see chapter 5 for more on inverse solutions). Therefore, the assumptions of the Sidak correction are not met in practice for neuroimaging.

The number of possible simultaneous hypothesis tests in neuroimaging modalities can be extremely large. In MEG data with 8,000 sensor space dipole locations, the entire phase locking functional connectivity network requires the simultaneous testing of approximately 32 million hypothesis tests. For the more conservative Bonferroni correction at the vertex level, the per comparison critical value would be  $\alpha = 3.9 \times 10^{-10}$ . In practice, this effectively eliminates all positive results and generally trades false positives in favor of a larger population of false negatives.

Alternatives exist for controlling the FWER. Importantly however, violations of the individual tests assumptions can result in FWER over or under estimations. One category of methods, exact tests, is a statistical test in which all of the assumptions of the underlying distribution of the test statistic are met. In contrast, approximate tests, such as parametric methods like the t-test, require that an approximation of the underlying parametric distribution be met by increasing the sample size used in testing.

The permutation test belongs to the category of exact tests and uses a non-parametric approach to generate an underlying distribution of the test statistic using the data observed from the experiment [51, 52]. This distribution is calculated by permuting through every possible way in which the observed data can be arranged between the test groups. A critical value is again chosen *a priori* and the observed test statistic is compared to the distribution of the permutation test. We reject the null hypothesis at the given significance level if the observed test statistic is less than that significance level. This type of control, of the per comparison significance level, allows the experimenter to derive a corrected critical value containing all the assumptions of the test.

As an example of the permutation test, suppose we have two groups A and B each with  $N_A$  and  $N_B$  samples respectively. We wish to test the null hypothesis that the samples from the two groups came from the same population (i.e. that the two groups are equal). We choose a test statistic and used permutations to generate the underlying distribution of that test statistic, and in this example we pick the difference in means between the populations. Our observed statistic is then  $\mu_A - \mu_B$  where  $\mu_A$  is the sample mean of the observations of A and  $\mu_B$  is the sample mean of the observations of B.

Next, we calculate the underlying population distribution of our chosen statistic by pooling all samples from the two groups in one group. We permute every possible set in which we can derive two groups with lengths,  $N_A$  and  $N_B$  without replacement. For each of those sets, we recalculate the mean of the two subsets and calculate our test statistic on this permutation by differencing their means.

The new distribution is then composed of all the calculated test statistics in each permutation. We want to assess whether the samples from the two groups came from the same populations. We calculate the probability distribution function (PDF) of the observed distribution via a histogram of the distribution of test statistics. Next, we calculate the two tail critical value empirically by finding the absolute value of the test statistic that contains the critical values' cutoff (i.e. for a critical value of .05, find the absolute value which corresponds to the tails of the distribution containing 5% of the PDF, or alternatively, the value which corresponds to 95% of the middle of the population). We reject the null hypothesis if the observed test statistic lies within the critical value tails of the underlying distribution. If it does not, then we fail to reject the null hypothesis.

Permutation testing offers advantages over traditional statistical tests. Most importantly, it requires no prior knowledge of the underlying distribution of the test statistic. It is calculated empirically given the observed data. Parametric statistics in contrast, such as t-tests, have specific assumptions that must be met with the observed data. Those assumptions must be tested before the parametric test can be performed. Violations of those underlying assumptions invalidate the test results. Furthermore, a permutation exists for any test statistic whether or not the underlying distribution is known. Again, this is due to the implicit calculation of the distribution of the underlying test statistic. For phase locking, this has the advantage of not requiring the assumptions of parametric models of the underlying test. Issues associated with the spatial correlation create dependencies between accompanying regions that, if not explicitly corrected for in parametric methods, would result in theoretical and practical FWER divergence as the result of violating the underlying assumptions of the test statistic. Permuting those dependencies and calculating the exact underlying distribution mitigates the responsibility of *a priori* knowledge or enforcement of the parametric distribution.

In addition, the use of permutation testing provides an exact p-value. That is, the assumptions of the p-value are met exactly. In parametric Gaussian statistical testing, the p-value is only approximate given that the data is only an approximation using the observed values of the theoretical data model. This results from the approximation inherent in many common parametric testing that the observed values are a sufficiently large enough sample size.

The major limitation of permutation testing is its computational complexity. For Gaussian parametric tests, the entire distribution can be characterized in two parameters the mean and the standard deviation (i.e. it is parameterized by those values). For non-parametric tests, no such parameterization techniques can be employed to reduce complexity. Additionally, for large  $N_A$

and  $N_B$  the required permutations may become excessively large to fully compute the underlying distribution. Therefore, the calculation of permutation tests requires significant computational power to calculate those underlying distribution and is many times more computationally expensive than for similar parametric sample sizes.

Finally, exact tests are less statistically powerful than parametric methods given the additional degrees of freedom in terms of correctly identifying true positives and minimizing false positives. A parametric test in which the given data is a close approximation of the expected data model, will always be more powerful than applying a non-parametric test to the same data.

### **3.4.2 Cluster Network Statistics**

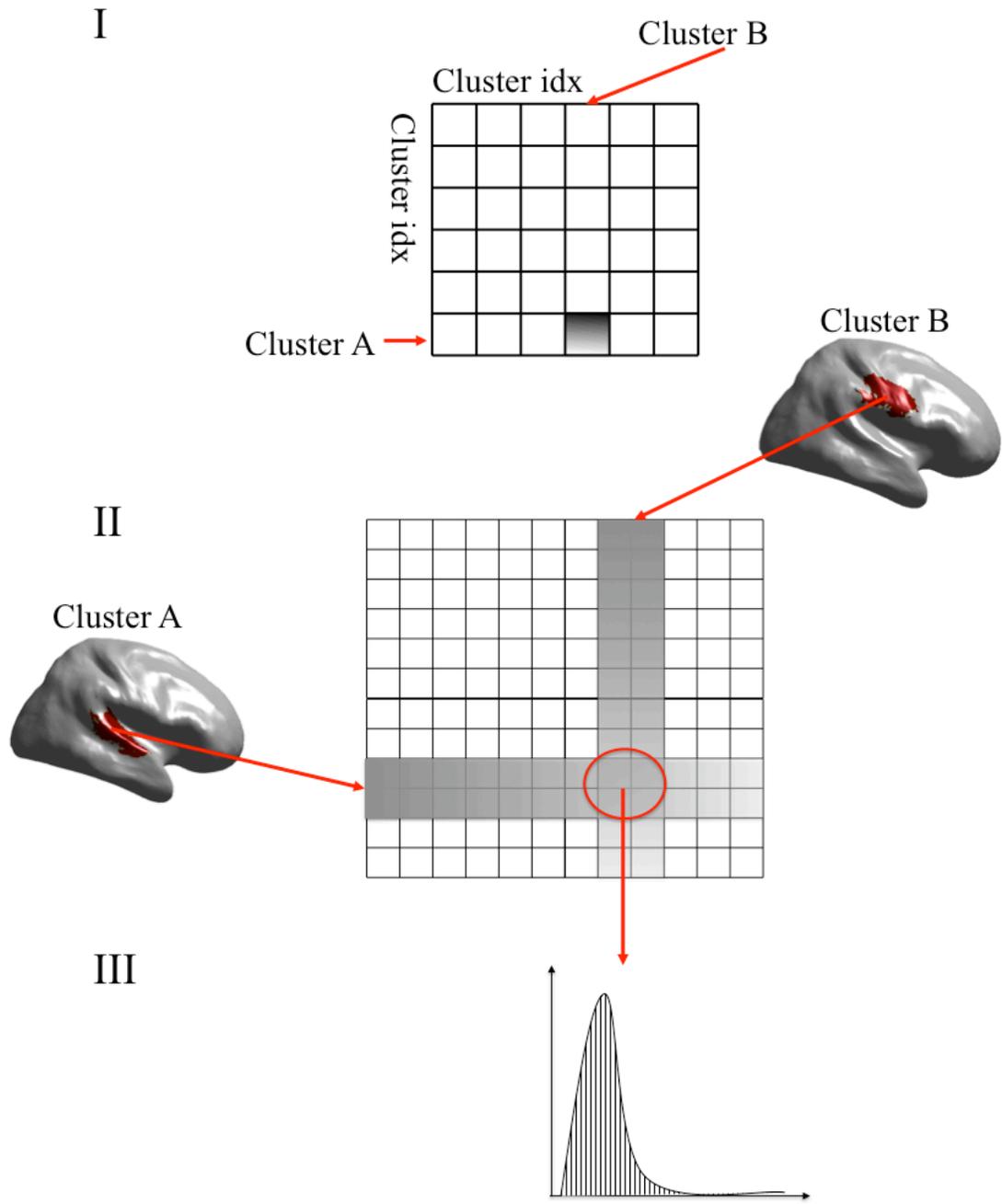
Permutation testing methods were applied to cluster level functionally defined regions using the phase locking graph data. Using the cluster level regions reduced the complexity of the patterns from roughly 33 million for 8,000 vertices to  $(2*k)((2*k)-1)/2$  (one  $k$  group of clusters per hemisphere with a typical value of  $k=40$ ).

Due to the large number of phase locking values calculated as well as spatial correlations between nearby vertices giving rise to high numbers of false positives, it was necessary to develop a robust method for the estimation of statistically significant phase locking graphs versus a null dataset using a permutation test. The null hypothesis was a phase locking graph containing equivalent cortical vertex locations but absent of any neurological activity (i.e. same meta-data coordinates). In chapter 5, we will make a concrete example of the null graph used in phase locking.

We are testing whether phase locking graphs from subject neurological data are statistically different than the connections in the null graph that is absent any neurological activity.

The test proceeds for each cluster pair  $C_1$  and  $C_2$ . First, for all vertices from  $C_1$ , we locate the set of edges that are incident upon vertices in  $C_2$  (including empty edges, i.e. zero values, so every possible way of connecting the two clusters in a complete graph). Our null hypothesis is that this set of phase locking values between the clusters  $C_1$  and  $C_2$  is not statistically different than those between equivalent data from  $C_1$  and  $C_2$  in the null datasets. In the adjacency matrix, this corresponds to the intersection between the cluster vertices.

This collection procedure was repeated for every cluster pair after accounting for symmetry between clusters (i.e. the distribution between  $C_1$  and  $C_2$  is equivalent to the distribution between  $C_2$  and  $C_1$  and is therefore redundant). In our experimental procedure, we include group level information in which we have  $r$  datasets corresponding to  $r$  subjects and  $n$  null datasets. The  $r$  subject datasets between cluster  $C_1$  and  $C_2$  were pooled into a larger distribution and similarly pooled for the  $n$  null datasets. These formed two groups each with their own distributions.



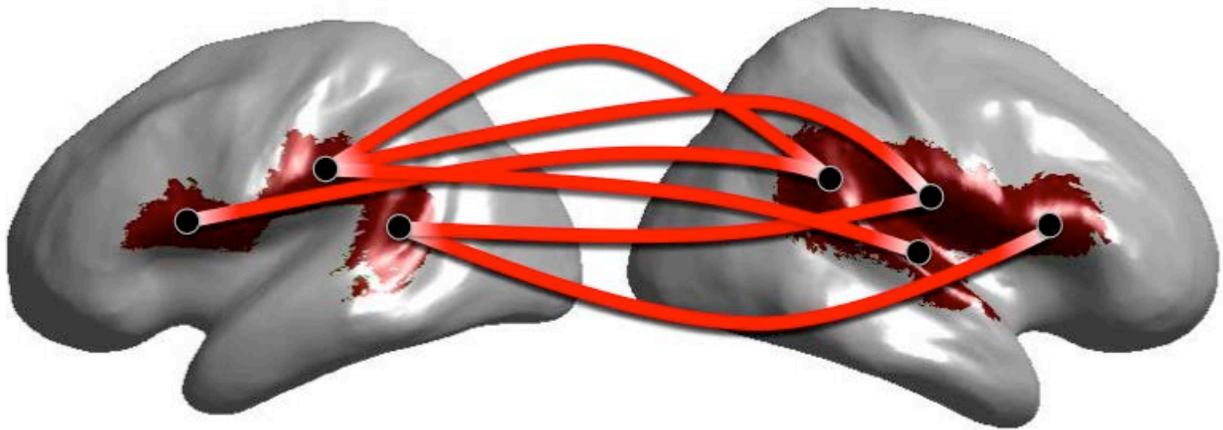
**Figure 3-7 Permutation testing on Clusters.**

Cluster indexes correspond to row or column indices in Figure 3-7 (I). For permutation testing on two clusters, the intersection point in Figure 3-7 (I) is used. Each of the clusters is then decomposed into the vertices in the phase locking graph adjacency matrix in Figure 3-7 (II) which are displayed as the grey bars. The overlaps in Figure 3-7 (II) of the two sets of vertices are the phase locking values that correspond to the two original clusters whose distribution is shown in Figure 3-7 (III).

We perform a permutation test between the global real set and null set. The procedure followed by permuting the combinations of the two populations. However, there were an excessive number of permutations possible for calculation within any reasonable computational limit. Some populations contained in excess of 30,000 cortical locations in a given cluster pair (high numbers are the result of subject registration, see section 6.2.3). The number of permutations is the factorial of the number of samples. Factorials are highly non-linear and for relatively small numbers of  $n$  can greatly exceed any reasonable computational time for all permutations (i.e.  $100! = 9.3 \times 10^{157}$ ). Therefore, a Monte Carlo resampling was used which is asymptotically equivalent for the permutation test [53, 54]. Using a method employed from [55], we created stopping criterion based upon the stability of the estimate of the underlying population's convergence. Namely, that the critical value estimation converges such that it has zero or has very small perturbations with successive iteration.

In practice, this number was small compared to the size of the population (statistics for number of iterations: average: 1132.5, standard deviation: 162.3. A minimum of 1000 iterations was enforced). Monte Carlo simulations were carried out for all cluster pairs by aggregating the resulting populations of  $r$  subjects across the total number of cluster pairs. The critical value for individual comparisons was then used to control the FWER using the global distribution of phase

locking statistics by correcting for the number of cluster level statistics. Using Monte Carlo permutation tests results in non-exact p-values, unlike the complete permutation test described above.



**Figure 3-8 A representative network identified in subject population**

Figure 3-8 presents a representative network at the alpha band frequency (8-13 Hz) showing the statistically significant connections between functionally defined regions across hemispheres. With this methodology, we are able to characterize the networks that exist in different frequency bands.

For our test statistic we used an unpaired t-test. To improve the reliability of the t-test the Rayleigh distribution of phase locking was transformed to a normal distribution. Rayleigh distributions can be pseudo-normalized by squaring the population values which results in a population consisting of the sum of two squared Gaussian distributions.

Finally, significant  $C_1$  and  $C_2$  connections are as those values whose observed test statistic is greater than the globally corrected critical value. Significant cluster interactions were visualized as a graph adjacency matrix of cluster vertices with significant permutation tests between clusters identified as binary edges.

### 3.5 CONCLUSION

The application of graph theory to functional connectivity analysis allows for the quantitative characterization and comparison of neurophysiological networks. For large scale processing, the implications of simultaneous hypothesis testing can create inconsistencies in statistical analysis. The multiple comparison problem is addressed explicitly within the network analysis methods described above. In addition, the data reduction techniques which utilize centrality measures to discover functionally defined seed regions further mitigates multiple comparison problems by drastically reducing the number of hypothesis under test without sacrificing network dynamics.

## **4.0 LEGION: DISTRIBUTED COMPUTING**

In this chapter, we describe a custom software library, Legion, which was written to perform distributed parallel computation of functional connectivity estimates in a time efficient manner. Phase locking is a computationally intensive algorithm on neuroimaging data sets and quickly scales to be computationally intractable. By utilizing clusters of computers that execute parts of the problem in parallel, the computational speed can be greatly improved.

### **4.1 INTRODUCTION**

Phase locking is a computationally expensive calculation between two time series. The calculation of instantaneous phase, followed by the averaging of differences is linearly complex and scales proportionally with the number of time samples and by the square of the number of locations. Neuroimaging techniques produce large quantities of data in spatial and temporal dimensions and thus the time series calculation rapidly becomes intractable. Application of functional connectivity to large quantities of data is not trivial due to this computational complexity and therefore necessitates parallel computational strategies to efficiently produce results within reasonable time frames.

Distributed computing makes whole brain functional connectivity computationally accessible by parallelizing independent computations across multiple computers that are then executed simultaneously. Functional connectivity in general, and phase locking specifically, is a single program multiple data (SPMD) computational paradigm in which the same computation is performed repeatedly over different independent data sets. In this programming paradigm individual computations require only the knowledge of the two time series being calculated which makes subsequent analysis independent. Therefore, we can parallelize the problem by dividing computations across multiple computers to be calculated simultaneously instead of serially. This results in a greatly improved computation time that improves as a function of the number of computers. However, this parallelization is trivial in theory but complex in practice. Practical implementation of large scale distributed computing frameworks requires management of multiple computers working towards a directed goal.

Here we describe Legion, a custom software package written to perform distributed parallel computation of functional connectivity estimates. The system is built upon a batch computing architecture and includes specific implementations for handling the pairwise computational nature of functional connectivity. It also provides the necessary mechanisms for aggregating and reintegrating disparate components of the processing mechanism into a cohesive answer. The following is a description of the overall theory of distributed parallel computation followed by the software architecture used for processing.

### 4.1.1 Distributed Computing

Distributed computing involves the use of multiple computers communicating over a computer network. Computer clusters are a server-based method of creating distributed computing hardware configurations. It involves connecting multiple server-grade computers over a low-latency network connection to allow faster inter-computer communication.

Distributed computing architectures can be setup in multiple configurations depending upon various performance considerations. Legion relies upon a master-worker configuration. In this configuration, a single master computer communicates with multiple worker computers. Workers conduct the real processing and receive instructions from the master. Workers also relay information regarding progress and job statuses to the master. The master is responsible for distributing jobs across the workers as well as monitoring, coordinating, and maintaining the overall status of the cluster.

In a batch computing system, specific workloads are called jobs that are submitted to the master for processing on a worker node in the cluster. The master accepts and queues the job in a first in first out (FIFO) stack. When cluster resources are available, the earliest submitted job is submitted to the worker that has become available and processing begins.

Batch computing frameworks offer a number of advantages over direct submission of jobs to computers in the cluster. Primarily, the master is capable of managing the cluster resources intelligently. It monitors the cluster resource utilization on each worker and submits jobs when and where processing resources are available. Additionally, batch-computing frameworks are more easily scalable horizontally (scaling out by adding additional workers). The addition of worker computers can be added to the available cluster resources allowing more jobs to be processed simultaneously. Likewise, worker nodes can be removed easily allowing for both downward scalability (reduction in cluster size) or for vertical scaling (scaling up by upgrading existing worker nodes to more powerful hardware).

Distributed batch systems have limitations. Jobs within this framework are restricted to the resources assigned by the master. Inter-job communication is not possible in our architecture's implementation. Each job must be a self-contained computational unit.

In addition, resource sharing in distributed systems is complex. Atomic operations become difficult where changes to one resource are immediately seen by every other resource. It often occurs in the situation in which two processes see different information regarding a resource's state. This can happen, for example, where one process writes a file and simultaneously another process attempts to read that file.

Sharing state information between the master and workers is typically via the master informing the worker of the current state. However, while processing a job, worker states can be affected by resource sharing if two jobs are running on a single multithreaded CPU. This leads to the sharing of the same CPU, RAM and disk resources amongst processor threads that can produce contentions. Special care must be taken when using distributed environments to prevent resource contention or hoarding between processes.

### 4.1.2 Computing Environment

Legion is built upon the specific hardware and software libraries available during development. It uses the grid computing architecture provided by Oracle Corporation called the Sun Grid Engine (SGE) for cluster management. SGE runs the portable batch system (PBS) that provides a resource-managing queuing mechanism that is accessed via a queue-submission protocol. SGE provides a series of tools for monitoring the cluster's current resource utilization, job distribution across the cluster, currently queued jobs and for maintaining user-specific information regarding who can submit jobs to which resources.

Each node in the cluster was installed with the MATLAB (Mathworks Natick, MA) computing software which is a JAVA based software package containing many commonly used methods in engineering and science. Legion job submissions execute a MATLAB instance on the worker that is used to execute the desired workload.

Logging and monitoring are derived in part from the SGE system in which the MATLAB console output is logged into a file. In addition, Legion logs job information to the Linux distributed logging facility rsyslog. This log provides cluster wide information regarding job progress.

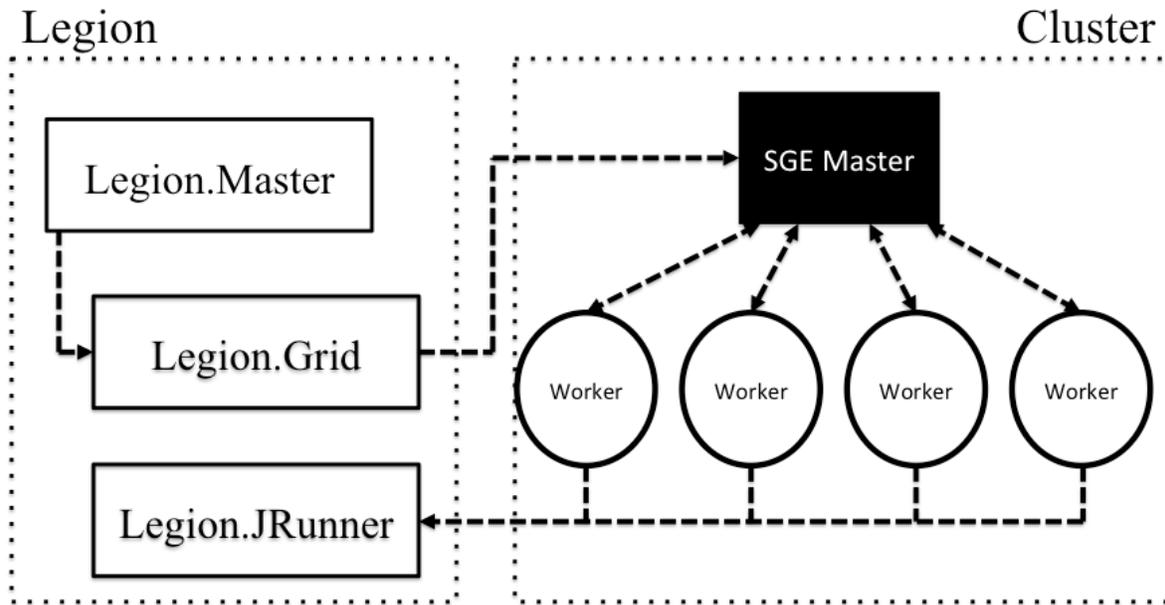
A shared network disk is used for dissemination of data loads from master to work processes upon job initiation. The shared disk allows outputs to be aggregated across the network and read back after completion of tasks. Our system uses a networked file system (NSF) and shares physical disks from the master over the private cluster network.

## 4.2 SOFTWARE ARCHITECTURE

### 4.2.1 Kernel Functions

Kernels are the Legion mechanism for providing function handles as first-class data types within MATLAB. It is similar to the built-in function handle system but provides a user defined arbitrarily complex pipeline operation similar to an anonymous function. A pipeline consists of an input data type as a single variable (structures, cells etc. allow for more complex input types) that is processed through N arbitrary functions. The function contracts require that a single output be emitted. Each output is subsequently carried into the next function argument as a parameter.

Kernel objects allow Legion to maintain a single interface definition within the Legion library while allowing user-defined customization. To obtain a blank kernel, we initialize it with the following code:`ker = legion.Kernel();` which accesses the default constructor for the Kernel class. To add arbitrary functions, we provide function handles followed by an argument list. The execution order within the internal pipeline is the same as the order in which functions are added. The add method has the following signature: `add( function_handle, arg1, arg2, ... );` where `add()` accepts as the first argument a MATLAB function handle. The argument list can be arbitrarily long, however, it must at a minimum contain a single 'X' flag which is used to represent where the output of the preceding function output should be carried into the current execution context, or in the case of the first method in the pipeline, the initial data set.



**Figure 4-1 Legion Architecture.**

Figure 4-1 shows a demonstration of the cluster, right, and how the software architecture interacts with the cluster, left. Legion has three general levels, the master where jobs are submitted. Legion Grid then contacts the cluster's master, SGE Master, for submission of the job to the cluster. When the SGE Master has resources available for computing the job it is sent to a worker process. Workers accept the job specifications and execute a MATLAB instance that runs the Legion Jobrunner needed for processing.

#### **4.2.2 Processing Components - Grid**

Legion provides an abstraction of the batch submission system while Grid takes care of the cluster-level details and handles implementation of job submission protocols to the underlying cluster. Grid is an extensible class within the Legion framework. Currently, Grid implements and fully supports the SGE backend running PBS for batch level processing. In this system, each node of the cluster framework is licensed to run a MATLAB instance. PBS accepts job submissions in a queued batch-system that distributes work to the cluster nodes based on a priority queue and resource management framework.

### 4.2.3 Master

The Legion Master class is the primary layer for submitting jobs via the MATLAB command line onto the cluster. It manages the consistency checks of the input data and accepts the kernel operation to be performed by each worker. The master is executed via the command: *master = legion.Master( kernel, data, numcores, jobName )* where the kernel is a Legion Kernel object, data is number of vertex data rows by data in either a cell or matrix format, numcores is the number of individual threads to execute on the cluster and jobName is a string which is used as the log location for the job output files. Note that the numcores must be less than or equal to the number of data rows and that the jobName must be unique to prevent data overwriting. This will execute onto the distributed processing system immediately. It will first initialize the input data and save it as one MAT file per numcores to the shared disk since this allows for simultaneous read operations without contention (within the limits of NFS). Then it will execute (using Grid for SGE/PBS) one pbs file per the number of cores requested to operate on each saved input data file. These jobs are queued in PBS where resource allocation of the cluster is managed.

#### 4.2.4 Jobrunner

Jobrunner is an internal class of the legion system that is responsible for the application of the kernel operation within a worker thread on the assigned data input. The input data can be any subset of the original data vertex. Jobrunner first loads its assigned input data, and executes a loop over the number of data rows in which each iteration initializes the provided kernel and saves the output of the kernel to disk.

### 4.3 STREAMING API

The streaming application programming interface (API) is an extension to the Legion execution system that implements a different data management strategy allowing for multiple batch computations and calculation of pairwise input data.

The streaming API introduces a preprocessing step prior to the master execution. First, data to be processed is written in blocks to disk where blocks consist of subsets of independent data inputs. Each output is emitted as a split file that provides both the raw data as well as information for reconstructing the original data input. The following function is provided:

```
legion.stream.Util.write_split_data( data, output_path, totalNumberSplits );
```

where the data is number of independent data rows by data, the output path is a string containing a path relative to the logs directory, and the totalNumberSplits is the number of splits for the input data.

### 4.3.1 Optimization and Customization Of Streaming API

The major advantage of the streaming API is the usage of a writer allows the data to be reused. Furthermore, the output of JobRunner is the same format as the input blocks. This provides a mechanism for cascading batch jobs. Batch jobs can be executed via the following:

```
master = legion.stream.Master(1);  
master.setInputPath( input_path );  
master.setOutputPath( output_path );  
master.setThreadKernel( kernel );  
master.setGridWorkingDirectory( jobName );  
master.setNumThreads( NumProcs );  
master.submit_job();
```

where we obtain an instance of a streaming Master, we set both the input and output paths relative to the log directory, specify the kernel, and specify a unique jobName as well as a number of processor threads to execute and finally submit the job to the cluster.

In addition, the streaming API also provides a number of additional customizations by specifying kernels for various processing stages. This allows even greater customization and optimization of processing. The following methods provide access to these extensions and require Legion Kernel inputs:

```
master.setSaveKernel( save_kern );  
master.setReadKernel( read_kern );
```

where the save kernel is executed when the JRunner finishes and saves the output of the thread kernel. The default is to save the output as a split file. The read kernel is executed on the split file when the JRunner process reads it. This allows for block level and row level processing for customization.

### 4.3.2 Input Pair-Wise Execution

The streaming APIs provide pair-wise calculation of each input thread. To calculate pair-wise arguments use the following master initialization with `master = legion.stream.Master(2)`; as shown in Figure 4-2.

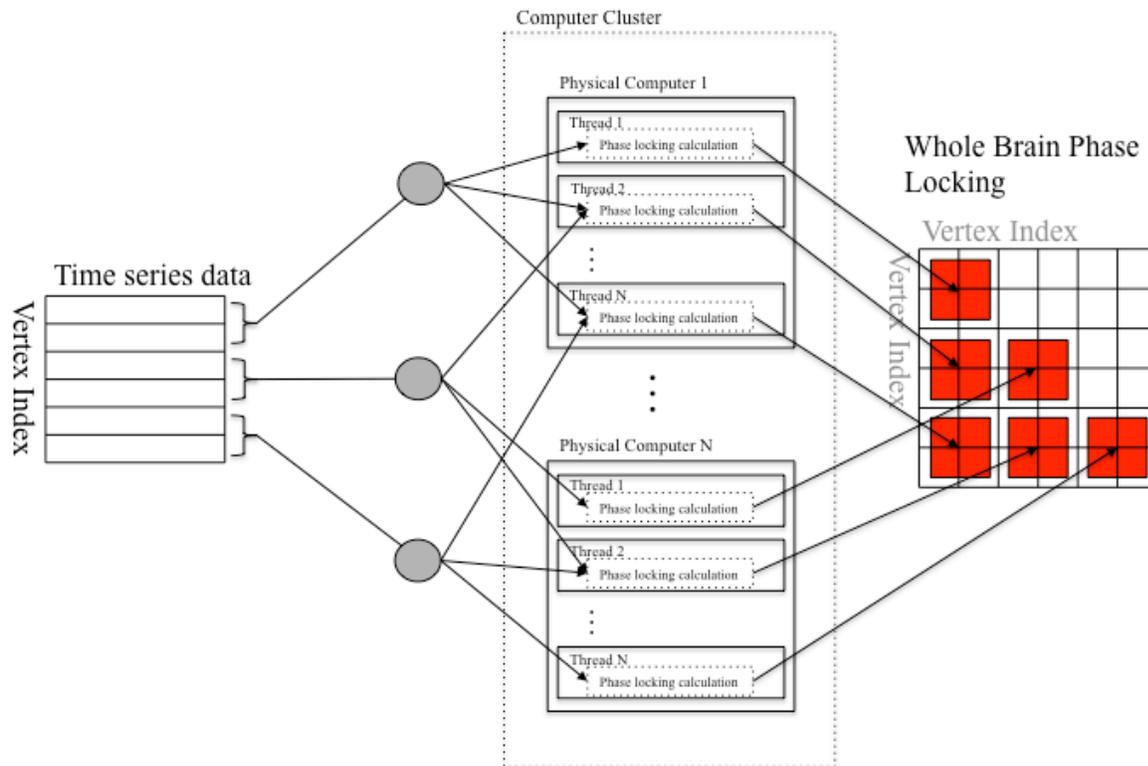


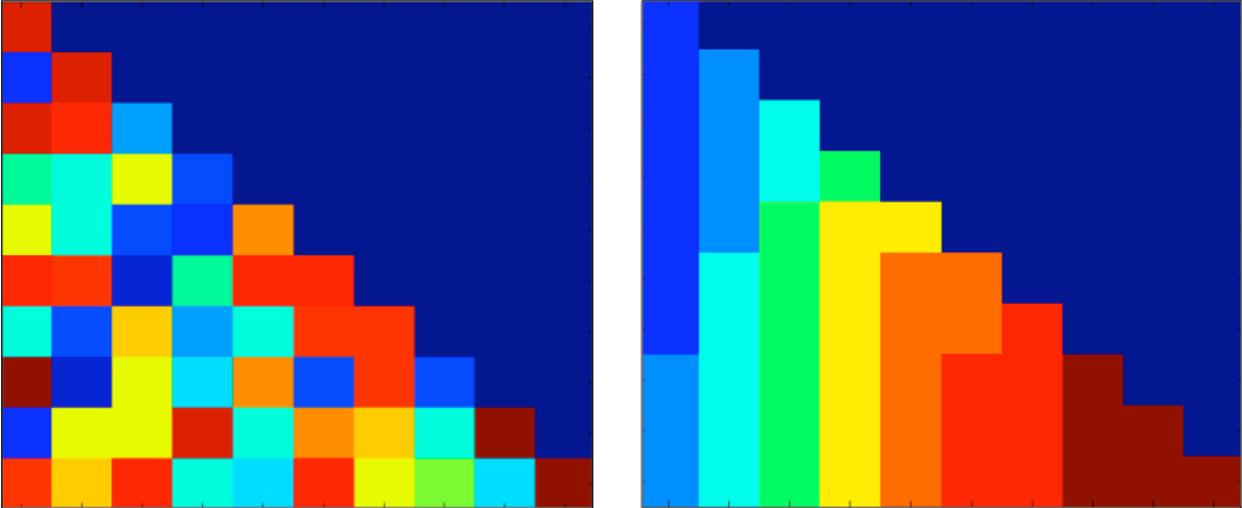
Figure 4-2 Legion Pairwise Computation.

Vertex by time series matrices are submitted as input to Legion for pairwise processing. The number of blocks is specified as the number of distinct breaks in the original input data's vertices, which above is three corresponding to three grey circles. The computer cluster's workers are physically separated into individual servers which each have multiple CPU cores each able to execute N threads. Pairs of blocks are distributed as distinct jobs to each worker thread (above there are 6 pairs of 3 blocks). Outputs from each worker thread consist of partial completion of the vertex by vertex output, on the right. Each red block indicates the portion covered by a single worker thread. Customization of the number of blocks and number of threads allows for fine-grained performance customization.

Where the (2) parameter value indicates the dimensionality of the processing (i.e. 1 for one per row, and 2 for pairwise). The JRunner class will then provide, within each block, the [i,j] row pairs within that block as input to the kernel function. Each block split pair will be executed on the cluster. For N block splits,  $(N+1)N/2$  block pairs will be executed.

The streaming API provides thread level control of the number of processors. It uses a deterministic hash (e.g. MD5) to assign jobs to workers. This deterministic hash is calculable on each worker such that workers can independently distribute jobs evenly amongst themselves without inter-process communication. Figure 4-3 shows two hashing results with the left indicating a random block assignment amongst workers and the right indicating a linear indexing along the columns such that I/O operations are minimized by repeatedly using the same input block and reading only a single block for each pair.

The streaming API determines the processing size and therefore memory footprint of each thread according to the split sizes. We can directly manage the memory required by each thread by changing the writer to different block sizes. For large jobs, the memory requirements can be the largest factor affecting performance especially on contended clusters.



**Figure 4-3 Block distribution mechanisms for distributed computation.**

To calculate the overall distribution of jobs per worker each worker thread uses a hash to determine the block pairs to calculate without duplication of effort. Left is a result of MD5 hashing of the input which randomly distributed pairs to each worker. Right is a linear hash that minimizes the number of block reads. For example, the first worker (blue in top-right) reads the first block and then subsequently reads single blocks in each block level loop until completion. In contrast, the MD5 hash requires each pair of blocks to be read, in general, since there is no reuse of block vertices.

After computing pair-wise calculations, the default output will be split files consisting of block-by-block indexes. Utility methods are provided for reintegration of these smaller block files into a single output file.

#### 4.4 DISTRIBUTED IMPLEMENTATION OF PHASE LOCKING

Phase locking is a computationally intensive algorithm. The steps required must be performed over each time point and at each frequency of investigation. For a typical MEG experiment, we have approximately 5 minutes of data. Sampling rates for MEG are 1 kHz that is downsampled to 250 Hz and corresponds to 75,000 samples.

At 75,000 samples, the phase locking calculation between two processes takes approximately 0.0056 seconds. To calculate 8,000 dipole locations at a single frequency then requires approximately 2.07 days of computation. However, a theoretically perfect distribution of these jobs to be executed over 80 computer threads then requires 0.62 hours. In practice, there is overhead associated with the distribution which prevents the theoretical maximum.

While distributing the phase locking estimations across multiple computers is a trivial theoretical exercise, in practice there is a great deal of overhead and complexity. In addition, the memory requirements for managing large volumes of data can create contentions in the distributed calculations as threads vie for additional resources. This is also not directly assessed in the theoretical calculations and is a further source of discrepancy between the theoretical and practical speed up of distributed computing.

Using 200 block splits of 8,196 input vertices (~41 vertices per block) and requires the total calculation of 861 block pairs. These jobs were executed on 20 threads (each thread responsible for roughly 43 blocks yielded thread completion times of an average of 8.3 hours with a standard deviation of .45 hours. As a per frequency estimate this is on average 2.07 hours per frequency. The theoretical speed of 40 computers is 1.24 hours per frequency.

The number of splits and number of threads were manually optimized to guarantee no resource contention on the nodes and allow maximal loading of the cluster. The RAM memory footprint scales as the square of the number of frequencies under investigation and linearly with the number of data points. For MEG data, these settings were found to optimize memory such that each shared computer in the cluster could execute approximately 8 simultaneous thread operations without interference or memory swapping.

The number of input splits has an effect upon the cluster performance as well. Increasing the number of splits will increase the number of read operations that in a shared network disk increases the bandwidth utilization and can create lags in processing, as processes must wait for the data to transfer over the network before beginning processing. Decreasing the number of splits increases the RAM requirement for the worker. Again, this scales as the square of the number of vertices and linearly with the number of data samples. The total raw input data of 8,196 vertices with 75,000 samples requires roughly 5 GB of memory. Setting the number of splits to one will require this amount of memory to hold the block. However, intermediate processing, such as the wavelet convolution creates a data object that is the same size as the block but repeated for each frequency. Therefore, for the case of a single split and single frequency, this requires approximately 10 GB of RAM. Therefore the choice of parameters is extremely important to successful processing.

## 4.5 CONCLUSION

Legion is a MATLAB software framework for computing large-scale functional connectivity networks in a time efficient manner. Phase locking functional connectivity is a computationally intensive algorithm whose computation time scales rapidly upon application to neuroimaging datasets. By utilizing clusters of computers, instead of repeated serial computations on a single computer, Legion is capable of distributing those computations and solving pieces simultaneously. This improved processing speed can lead to the application of functional connectivity to whole brain functional connectivity networks. Large-scale data analysis provides a tool for neuroscientists to investigate the complex network dynamics of neural populations.

The Legion software library provides a generic software framework for batch computing using MATLAB. It provides the explicit capability for handling pairwise computations that are required for functional connectivity studies. It's distributed computing architecture greatly improves processing times making previously intractable or computationally prohibitive analyses accessible to a much broader range of applications.

### 4.5.1 Caveats and Limitations

The major challenges with legion are the same with any distributed batch system, namely that cluster resources are scarce. Over utilization of memory or disk IO can cause bottleneck issues that are difficult to debug. Further, PBS is not able to predict peak resource utilization in advance during long running processes that can cause issues if multiple jobs are allowed to expand without limits. This is especially pertinent given MATLAB's liberal use the Java's memory heap. In general, MATLAB expands both memory and CPU utilization to the limits of the

physical machine. Meaning that running multiple MATLAB instances on a single computer node can cause failures as they can each utilize the full computer hardware. In general Linux maintains this contention well but large data sizes or complex algorithms can quickly bring down a cluster. It is the responsibility of the program designer to ensure there is no over-utilization of cluster resources that is why so many optimization paths are provided to allow performance enhancements.

Finally, this is a distributed shared-disk system with consistent read and non-atomic write protections (same for any NFS backed system). Reading immutable data files can be relied upon for consistent state but may still result in blocking I/O operations over the network or on the physical disk. However, write operations on the same file pointer can cause thread synchronization issues between batch jobs. Jobs in which inter-thread communication are necessary cannot be computed at present. The easiest mechanism to support this paradigm would be to utilize a cascade of jobs. At present, this is untested and no reliable programmatic mechanism is supported for waiting until job completion before beginning the next cascade step. However, methods are provided for testing the current status so an extension could be created.

## 4.5.2 Future Work

A future extension to Legion is MapReduce. MapReduce is a computational paradigm for processing large datasets on clusters of computers. It consists of a map stage and a reduce stage. During map, input data is accepted as key value pairs and emitted after processing in each distributed process as a new key-value pair. During reduce, every equal key is processed on the same reduce process and aggregated across all possible maps that emitted that key. Each reduce then allows all the same key's list of values to be processed.

Legion was built similarly to the framework of the Hadoop implementation of MapReduce by maintaining consistent input and output file types in a key/value architecture. It lacks the unique key naming scheme and only allows integer values. Future work would include an implementation of the reduce sorting algorithm. That would allow two Legion jobs cascading legion jobs with a sort in between to recreate the MapReduce framework. With that implementation, MapReduce could be implemented in Legion and allow a wide range of processing methods with full access to all available MATLAB libraries.

## **5.0 WHOLE BRAIN PHASE LOCKING GRAPHS IN MAGNETOENCEPHALOGRAPHY**

In this chapter the previously established methods of phase locking graphs are applied to magnetoencephalography. Simulation experiments are conducted to verify the validity of the network analysis methods given *a priori* neural dynamics. In addition, investigations of the empty room phase locking graph were conducted.

### **5.1 INTRODUCTION**

Earlier chapters established phase locking functional connectivity methods for analysis of time series neurophysiological activity. In this chapter, those methods are applied to magnetoencephalography (MEG) neural signal records. MEG is a non-invasive measure of neural activity. It produces time series data recorded from whole brain neural activity measured at sensors located outside the head.

MEG sensor locations are external to the subject's scalp but contain contributions from multiple neural populations distributed throughout the volume of the brain. As a result, there is no unique solution for reconstructing the location of the neural activity. Established methods for estimating these locations produce spatial correlations in the cortical neural activity reconstructions that affect phase locking's resolution.

Furthermore, the complexity and scale of whole brain phase locking gives rise to spurious events that must be accounted for when performing statistical tests as a direct result of the multiple comparison problem. The use of an effective null data set, a phase locking graph absent neural activity but still subject to the induced biases of the reconstruction methods, is imperative for discerning true phase locking networks from spurious connections arising by chance alone.

Here we present methods for the direct application of phase locking graphs and the associated analysis for functional connectivity network detection at a given frequency band. The characteristics of the reconstruction method are investigated specifically towards their effects upon phase locking values distributed across the brain. To validate these methods for use in cortical neural network investigations, we use mathematical simulations for validation that the methods introduced previously yield results consistent with *a priori* knowledge of the underlying neural activity.

## 5.2 BACKGROUND

MEG operates by recording magnetic fields associated with neural activity. Synchronized neural activity in a localized region produce ionic currents which give rise to magnetic fields according to Maxwell's Equations[56-58]. More specifically, a flowing charged particle gives rise to a magnetic field orthogonal to the direction of travel. This process can be characterized by an electric dipole having a position, direction and magnitude. MEG records the net effect of these electric dipoles produced from neural activity.

The electric currents involved in MEG neuroimaging are extremely small and typically on the order of 10 femtotesla ( $1\text{fT} = 10^{-15}\text{ T}$ ) for cortical neural activity. The ambient magnetic field is on the order of  $10^8\text{ fT}$ , eight orders of magnitude higher than human cortical activity [59]. To detect these small magnetic fields, superconducting quantum interference (SQUID) devices are used. SQUIDs are extremely sensitive magnetometers capable of measuring magnetic fields as small as  $10^{-18}\text{T}$ .

In addition to Earth's magnetic field, other sources of magnetic interference are projected into the experiment environment. To reduce the contributions of external magnetic fields and thereby obtain maximal signal to noise ratio of neural signals, MEG data is collected within the confines of an electromagnetically shielded room. However, additional post-processing techniques are employed to further reduce noise not originating from within the brain (see section 6.2.1).



Figure 5-1 MEG System.

MEG (Figure 5-1) is recorded using an array of SQUIDs located external to the scalp (In our system, 306 sensors are used). Neural populations that produce ionic currents, give rise to magnetic fields that are recorded by multiple external SQUIDs in the array. Approximately 50,000 active neurons are needed to produce an electric dipole detectable by a SQUID [60]. To have a net summation giving rise to a strong magnetic field further necessitates that the active neurons involved be similarly oriented so that their magnetic fields positively sum. Pyramidal cells located within the cortex are oriented perpendicular to the cortical surface and their concurrent activity gives rise to a magnetic field strong enough to be detected by MEG and with an orientation necessary for detection by SQUIDs. Due to the folding of the cortical surface, neuronal populations of pyramidal cells located within sulci are oriented tangentially to the scalp and is therefore the location of MEG's greatest sensitivity. In , the left is an image of the door and its associated shielding. Right, a subject sitting in an MEG scanner with head near the sensors.

MEG is predominantly sensitive to intracellular currents. Dipole generation as the result of ion movement in extracellular spaces tend to cancel (such as the postsynaptic junction) due to interfering magnetic field generation and the cancellation of any induced dipole fields. Whereas currents produced intra-cellularly within dendrites have a net ion flow giving rise to a measureable magnetic field [61].

Deep brain structures are difficult to measure with MEG. For magnetic dipoles, the magnetic field strength decays according with a  $1/r^3$  parameter where  $r$  is the distance to the dipole. For increasing distances from the intracellular currents the magnitude of magnetic field reaching the SQUIDs is extremely low. Therefore, neural recordings within MEG are dominated by the superficial recordings of neural activity such as those within cortical structures.

Measurements in the sensor space of the SQUIDs are an ill-posed inverse problem in which the location of the original neural activity cannot be uniquely determined from observations in sensor space. Ill-posed problems occur because the number of source generators exceeds the number of locations from which to observe the phenomena. This problem is characteristic of surface measurements of a volume.

Methods exist which allow estimation of the original source locations that are improved by including prior information regarding the problem's configuration[56-58]. However, these methods can produce inconsistencies between the underlying truth and the reconstructed neural activity. The nature of those inconsistencies gives rise to artificial neural patterns that become important when interpreting the functional connectivity associated with the MEG data.

In the most general form, MEG cortically reconstructed signals arise from the active firing of neural populations giving rise to magnetic fields plus the addition of noise terms. Empty room data provides an estimate of the baseline MEG absent neural signaling by recording a similar paradigm to the active trial (with a subject brain) but without the presence of a real subject. Noise produced external to the sensor space covered by the MEG sensors will be present in empty room recordings (within realistic tolerances as the conditions will never be exactly equivalent) as well as arising from noise in the SQUIDs. Following reconstruction of the empty room sensor recordings to source space, the covariance structures imposed by the inverse solution will also be present.

## 5.3 METHODS

### 5.3.1 Cortical Surface Reconstruction

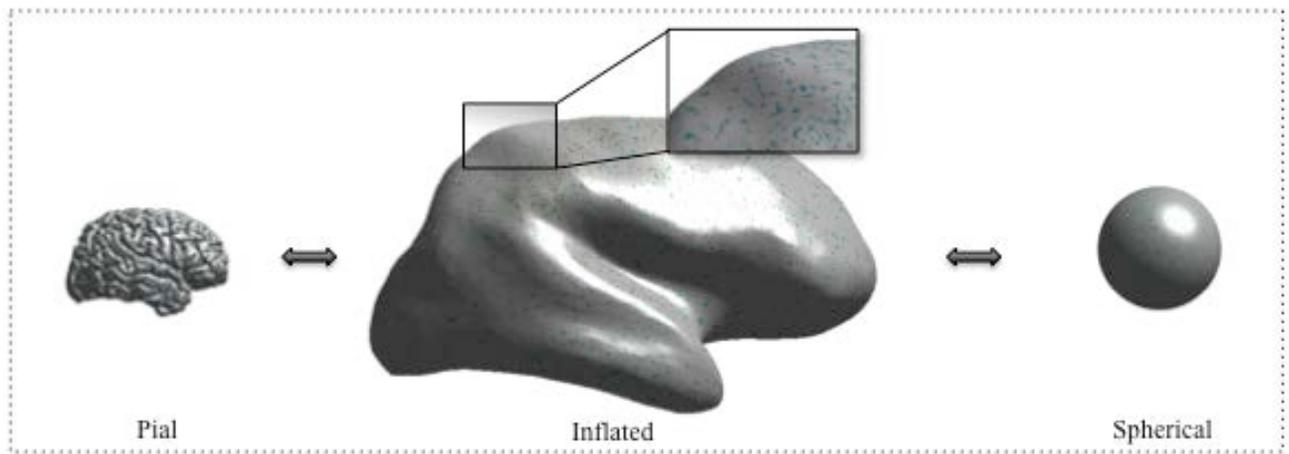
MEG data is recorded at sensor locations external to the scalp. Each location is a superposition of multiple source generations of magnetic fields within the brain assuming instantaneous magnetic fields. Physiologically interpretable data necessitates that results be provided at cortical locations rather than in sensor space. However, the ill posed nature of MEG reconstruction presents difficulties in discerning true neural signal locations. Therefore, spatial correlations are introduced along the cortical surface because spatially close cortical vertices in the inverse model formulation contain large contributions from similar sensor space recordings.

For phase locking these spatial correlations give rise to spurious relations between nearby vertices that are not necessarily from true neural signaling in adjacent cortical locations [2]. To mitigate these spurious correlations, we have utilized the empty room recordings to empirically estimate the spatial correlation of nearby vertices absent neural signals that arise from the inverse solution.

### 5.3.2 Minimum Norm Estimation of Cortical Locations

Reconstruction of cortical dipole locations proceeds by first characterizing the cortical anatomy of each subject. Each subject had an anatomical structural MRI image collected using a 3-Tesla whole body scanner. The Freesurfer™ software package was used to segment a cortical surface model from the structural MRI [90-95]. It creates a mesh consisting of approximately 150,000 vertices and the corresponding triangular faces connecting the vertices to form a cortical

surface representation. The subject specific cortical mesh was downsampled to contain 4,098 cortical surface vertices per hemisphere corresponding to an approximately 10 mm inter-vertex spacing along the cortical surface. These vertices were then used as dipole location estimates for the source reconstruction.



**Figure 5-2 Cortical Surface Dipole Locations.**

In Figure 5-2, the three cortical surface representations are transformable. Left is the pial surface that is representative of the anatomical cortical surface. Middle is the inflated brain that shows a smooth surface without gyri and sulci. Right is the transform of pial surface into spherical coordinates.

We estimated the cortical locations using the minimum norm estimator (MNE software suite, v2.7)[2, 56-58, 62-64]. The MNE is calculated by applying a linear inverse operator to project sensor space signals into the higher dimensional source space:

$$\hat{y}(t) = Wx(t) \quad (5-1)$$

where  $\hat{y}$  is a matrix of  $n_d$  dipole locations by time,  $t$ , and  $x(t)$  is a matrix of  $n_c$  sensor space recordings by time,  $t$ .  $W$  is the inverse operator that performs the projection operations between the two signal spaces. The MNE solves:

$$W = \arg \min_y \|C^{-1/2}(x - Ay)\|_2^2 + \lambda^2 \|R^{-1}y\|_2^2 \quad (5-2)$$

which can be minimized via:

$$W = RA^T(ARA^T + \lambda^2 C)^{-1} \quad (5-3)$$

where  $R$  is an  $n_d$  by  $n_d$  matrix denoting the covariance of the sources.  $C$  is an  $n_c$  by  $n_c$  matrix denoting the covariance of the noise in the sensor space.  $A$  is a  $n_d$  by  $3n_d$  matrix consisting of row vectors which correspond to the source orientation solution of the forward problem with each column denoting the orthogonal magnitude of the vector.  $\lambda$  is a regularization parameter to improve instabilities in the inverse solution and is calculated as:

$$\lambda = \frac{\text{trace}(ARA^T)}{\text{trace}(C) * SNR^2} \quad (5-4)$$

where a value of the SNR was chosen to be 3 consistent with previous MEG analysis. (Hämäläinen, M. MNE software user's guide version 2.7, 2009) The depth dependent decay of magnetic fields can be accounted for by applying a depth weighting to the components of  $A$ . From [64], the columns in  $A$  can be monotonically depth weighted via:

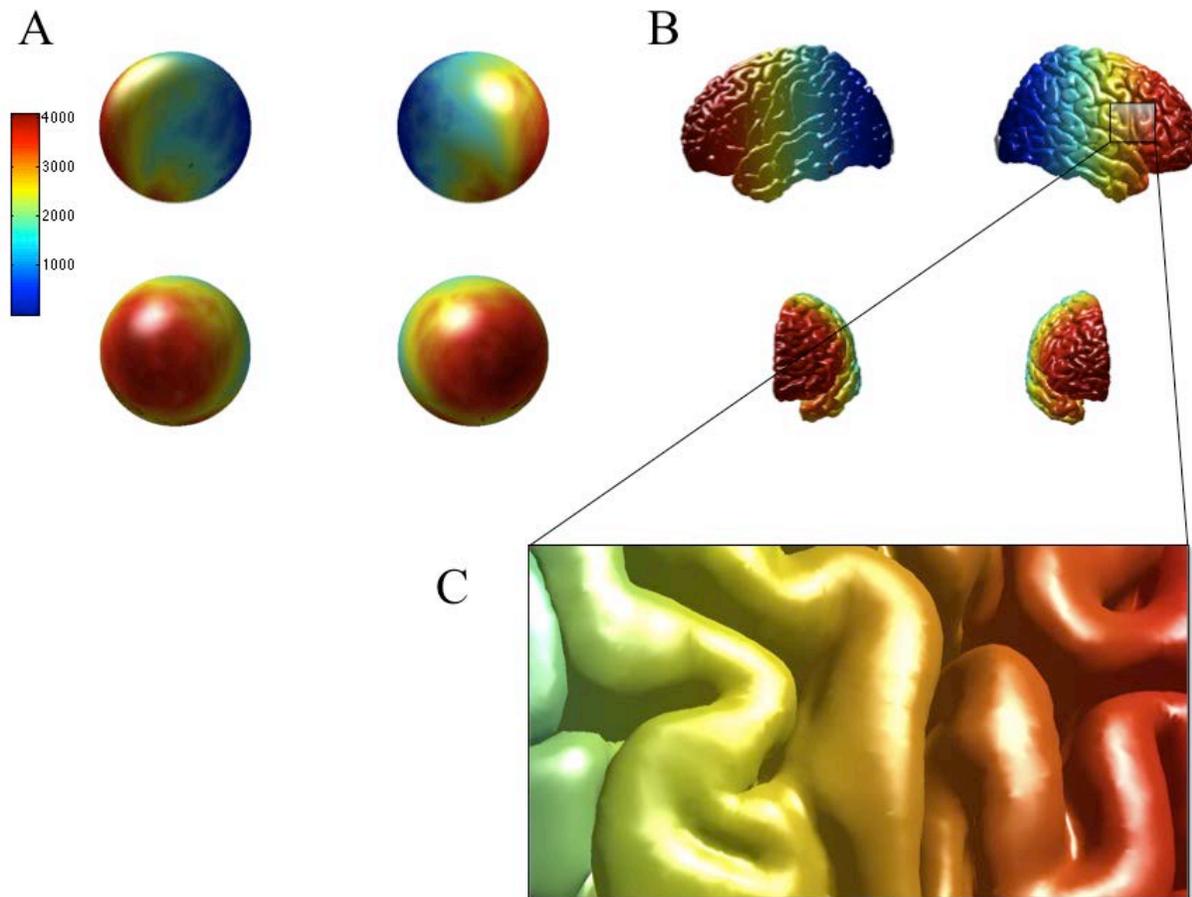
$$f_k = 1/(a_{3k-2}^T a_{3k-2} + a_{3k-1}^T a_{3k-1} + a_{3k}^T a_{3k})^\gamma \quad (5-5)$$

where  $a_p$  is the  $p^{\text{th}}$  column of  $A$  incrementing by the three columns per dipole location in the columns of  $A$  and  $k$  denotes the dipole index within  $A$ .  $\gamma$  is a tunable parameter that was set at .4 from previous MEG studies [2].

The covariance matrix  $C$  was calculated by using empty room recordings in sensor space. In stimulus based experiments,  $C$  would be calculated from rest periods with a subject present to account for covariances between sensor time series. However, these covariances arising from interactions between cortical vertices are of interest to our investigation. Therefore, including them in the preprocessing steps would remove the covariance between signals that we are trying to characterize.

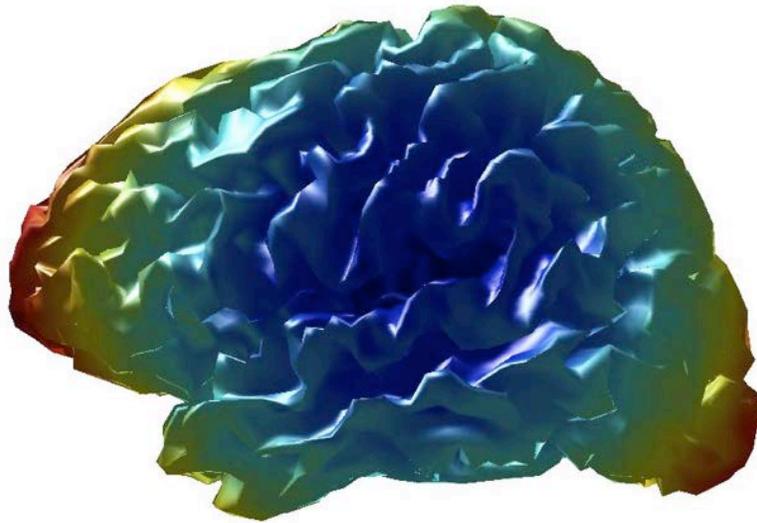
### **5.3.3 Properties of MEG Cortical Reconstruction**

Figure 5-3 plots increasing vertex numbering in the two hemispheres. Cooler colors (blue) indicate lower index values while warmer colors (red) are higher index values. Counting begins at the posterior cortical surface and proceeds in a spiral pattern to the anterior of the brain. Two cortical indexes are close in Euclidean space with respect to the spiral direction. However, the nearest neighbor cloud with respect to Euclidean surface distances will incorporate points occurring in previous revolutions or future revolutions. In this instance, the linear vertex numbering does not accurately represent the distance between dipole locations. The plots show similar results for the inflated surface in which the cortical surface distances can be seen more closely to not be radially close. This is an important result when interpreting adjacency matrices since one-dimensional indexing of the three-dimensional data yields discrepancies in vertex distance along the cortical surface.



**Figure 5-3 Vertex Numbering After Cortical Reconstruction.**

Figure 5-3 shows one-dimensional vertex ordering of the 2-dimensional surfaces. Numbering begins at the posterior base of the brain (blue) and continues in spirals towards the anterior (red). In (A) the spherical coordinates' vertex indices are shown. In (B) the pial surface vertex ordering are shown. The zoomed in section in (C) shows a close up of the gyri and sulci. Spiral indexing runs vertically, therefore, the nearest points can occur along horizontal directions. The 1-dimensional indexing is not explicitly sorted by nearest neighbors; however they are still roughly close.



**Figure 5-4 Average distances from vertices as a result of vertex numbering pattern.**

For each point, the distance from that vertex to every other vertex along the cortical surface was calculated. The average of those distances is displayed above. Inconsistencies in the distribution of average values result in biasing of phase locking values. Centrally located vertices are on average closer to all other vertices and vertices located at the anterior or posterior extremes have, on average, longer distances to every other vertex. The point spread function of MEG as a result of the minimum norm estimation of the surface dipole locations result in spatial correlations that are a function of distance. Central vertices with average shorter distances will have a larger false-positive cloud and consequently will dominate most centrality measures.

In Figure 5-4, we note that as a function of cortical surface location, the average cortical surface distance between all other locations is not uniformly distributed. Specifically, regions located at the anterior and posterior regions are on average connected to more distant points while regions located centrally have fewer long distance vertices and more long distance connections.

This has implications in the distribution of each vertex's point spread function that is highly dependent upon the distance between points (as a condition of the minimum norm estimation). Nearby points on the cortical surface will have higher contributions from the same sensor space recordings and therefore their source space activity will have a higher phase locking value. Centrality estimates will tend to bias those points which are highly connected to greater numbers of nearby high-strength vertices and therefore regions located centrally will have a disproportionate number of high strength connections.

### **5.3.4 MEG Simulation**

MEG simulations were conducted by simulating *a priori* dipole activity. This simulation procedure follows the simulation steps given in the MNE<sup>TM</sup> manual v2.6 for the `mne_simu` function [65]. Sensor space recordings can be generated using the forward solution model parameters by simulating the sensor recordings of that dipole activity. This simulates the theoretical observation of the simulated dipole locations. Using the above reconstruction methods, we can investigate the effect of the inverse solution on phase locking graphs relative to known dipole signals.

Vertex labels on the cortical surface can be simulated, one at a time, with an equation based signal. For phase locking sinusoids were used. Multiple vertex locations are superimposed by generating source space locations with varying signal generation techniques by aggregating multiple simulations.

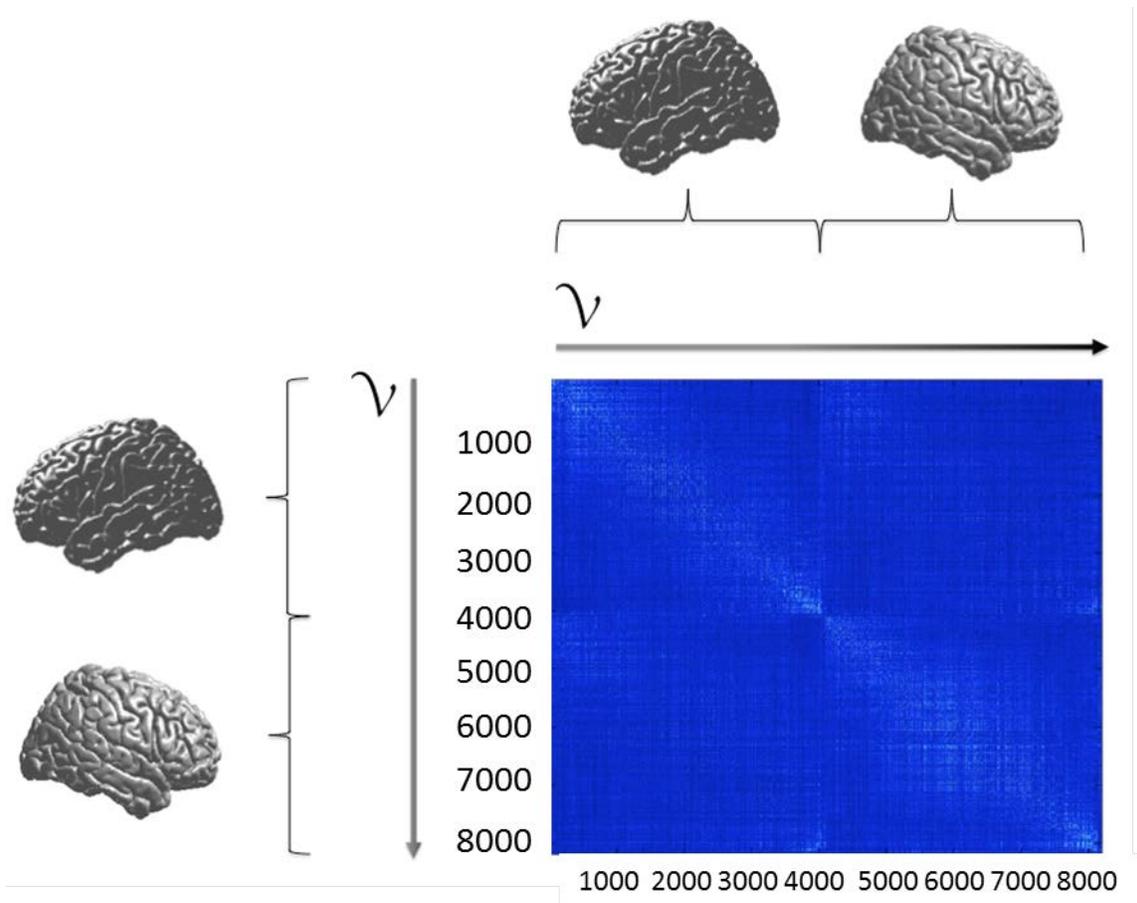
The simulation begins by first simulating each cortical vertex composed of the desired signal at certain vertices and then adding white noise at every dipole vertex location. The forward solution is applied to the cortical locations to obtain the sensor space recordings given the data. Next, the inverse solution is applied to the sensor space data in a similar manner for real data as above. This simulated time series represents the estimation of cortical location activity according to the constraints of the observed time series and the assumption of the inverse model. Additional noise can be added at each stage of the process to modify the signal to noise ratio of the simulation or test various assumptions of the model.

For MEG simulations the SNR of the resultant dipole generators is scaled such that the  $n_{ave}$  parameter to the `mne_simu` function increases the SNR inversely proportional to the square root of  $n_{ave}$ , where  $n_{ave}$  is the number of averages performed in the simulation. In our simulations the parameter `nave` was set to 1 for a minimum SNR. Alternatively, for higher SNR a value of either 10 or 100 was used depending upon the required SNR scaling. In addition, the following equation was used to generate the sinusoids at two vertex locations to generate a phase locked signal:  $q = 26e-9 * \sin(2 * \pi * 15 * x)$ , where  $x$  is the millisecond time index of the recording. Perfect sinusoids between two locations will be phase locked across all frequencies. However, the addition of Gaussian noise in the simulation creates phase shifts that reduce the phase locking preferentially at off frequencies resulting in phase locking only at the 15 Hz frequency of the original sinusoid.

### 5.3.5 Phase Locking Graphs in MEG

The phase locking graph methods in chapter 3 were applied to MEG signals following the cortical reconstruction procedure above. Whole brain phase locking graphs were created from 4096 dipole time series per hemisphere.

In Figure 5-5, vertices correspond to left-hemisphere followed by right hemisphere (1-4098 are left-hemisphere and 4099-9086 are right hemisphere). The lower left triangle and upper right triangle are mirrors given the symmetry of phase locking (i.e.  $PLV(A,B) = PLV(B,A)$ ). The top left quadrant of each phase graph is the intra-hemispheric connections of the left hemisphere with vertices within the left-hemisphere while the lower right quadrant is the intra-hemispheric connections of the right hemisphere. In both we see a diagonal banding which is a result of the spatial correlations in the inverse solution, namely that nearby cortical vertices have similar sensor-space signal contributions. The lower left quadrant is the inter-hemispheric connection and represents phase locking values originating in one hemisphere and terminating in the opposite hemisphere. We again see a diagonal element that originates from the interior surfaces of each hemisphere since cortical vertices between those cortical regions are close in Euclidean space despite being in different hemispheres. Again, these shorter distances are influenced by the spatial correlation patterns of the inverse solution.



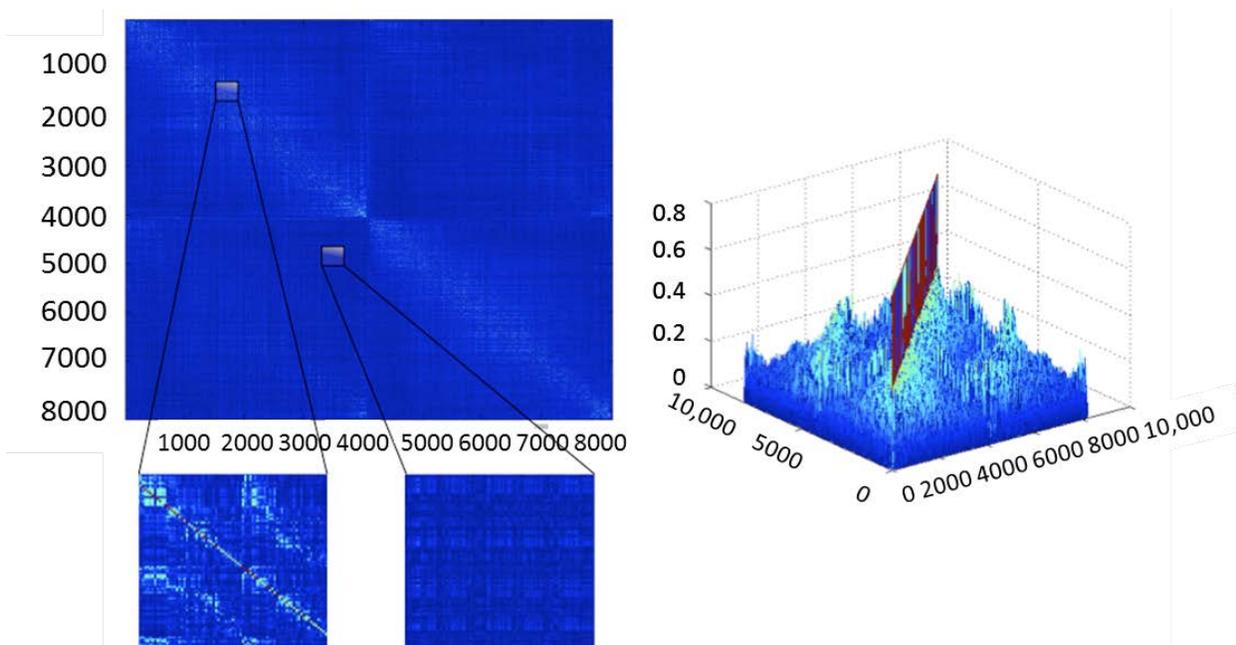
**Figure 5-5 MEG Phase Locking Graphs.**

Phase locking graphs consist of vertex-by-vertex matrices in which each entry is the phase locking value between the vertices corresponding to the row and column of that entry. The first half of the vertices belongs in the left hemisphere and the second half belong in the right hemisphere. Phase locking graphs are symmetric and are mirrored across the main diagonal. The top left quadrant consists of the intra-hemispheric connectivity of vertices in the left-hemisphere and those phase locking values that connect to other vertices also in the left hemisphere. The lower right quadrant is similar but consists of the intra-hemispheric right hemisphere. The lower left and upper right quadrants correspond to the inter-hemispheric connections in which one

vertex is in each hemisphere. Characteristic of MEG phase locking graphs, the above adjacency matrix shows a diagonal dependence as a result of nearby vertices having large false positive point spread functions.

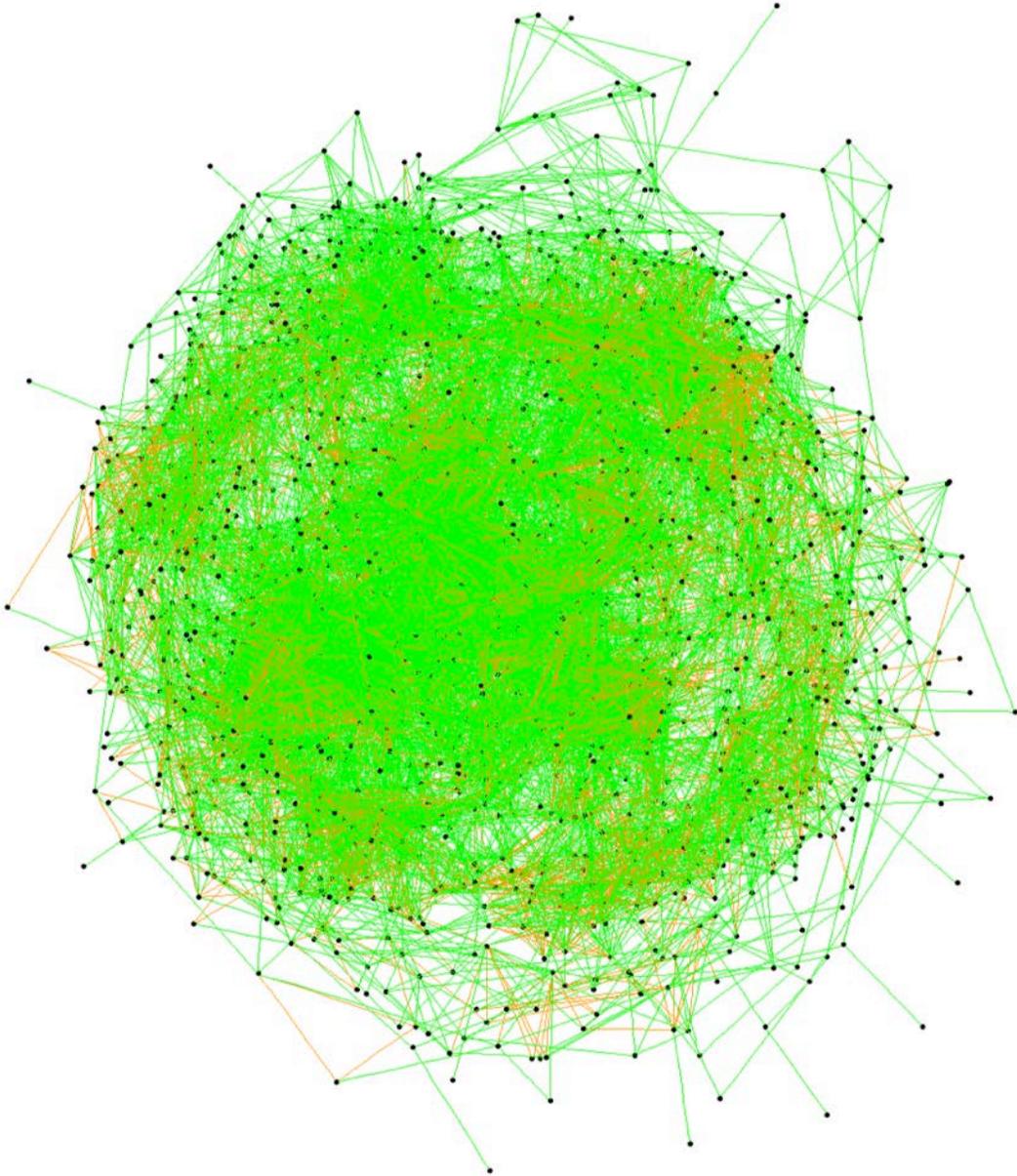
In Figure 5-6, we see increased resolution images of the phase locking graph. It should be noted that matrices of the size 8,192 by 8,192 cannot be visualized in full. Therefore, a considerable amount of structure is lost when visualizing these images. The smaller regions demonstrate the patterns present at higher resolutions. In the right of Figure 5-6, a mesh is plotted of the same data that allows easy visualization of the main diagonal and the inter-hemisphere diagonals. The non-smooth diagonals (inter- and intra-) are the result of noise in the calculation of the inverse solution as well as the indexing of the three-dimensional points in one-dimension.

Figure 5-7 is a graph of select vertices within a phase locking graph. To make visualization possible, only a quarter of the roughly 8,000 vertices are displayed. Black dots represent vertices while the color of edges depicts the strength of the phase locking connection between those vertices where green is less than 0.2, orange is less than 0.6 and red is greater than or equal to 0.6. Green predominates the graph as expected from Rayleigh distributions of phase locking values. Some points have relatively few connections as the result of the downsampling of the original vertex space; many points connect to nearly all vertices. In the full graph representation this effect would be more recognizable.



**Figure 5-6 Phase Locking Adjacency Matrix.**

Phase locking graphs consist of roughly 8,000 vertices with approximately 32 million phase locking values in the lower left triangle. Magnified portions of the phase locking value are shown in the bottom left from near the main diagonal (left) and inter-hemispheric phase locking values (right). On the far right, a mesh depiction of the phase locking graph. The red line down the diagonal corresponds to strong false positive spatial correlations at close vertices. This mesh is a mirror image across the main diagonal. In the inter-hemispheric quadrant, (far left and far right) there also exists a diagonal element seen running parallel to the main diagonal. This is a result of extremely close points in inter-hemispheric connections as a result of the indexing of the cortical surface.



**Figure 5-7 Representation of Phase Locking Graph.**

Black points correspond to vertices, and lines connecting them are edges of phase locking values between vertices. Edge color indicates the strength of the connection where green is less than .2, orange is less than .6 and red is greater than or equal to .6. Green connections predominant as expected from the Rayleigh distribution.

## 5.4 RESULTS

### 5.4.1 Empty Room Noise

In Figure 5-8, the Eigenvector centrality of an empty room phase locking graph is displayed on the cortical surface. The color of the cortex indicates the centrality scores at each vertex where warm colors represent high centrality and therefore high importance of that vertex within the graph while cool colors represent lower centrality and therefore lower importance of that vertex within the graph. Centrality scores are normalized to one. The centrality dominance within the central regions near the temporal lobe is the result of large numbers of nearby cortical locations compared with posterior and anterior regions of the brain. Also note that for the inflated brain surfaces, the high centrality regions present in the pial surface are no longer present as the calculation of the inflated surface tends to smooth the interior surfaces of temporal region.

In empty room data, there are no neural signals which creates a mean phase locking value of 0.15 (see section 2.4 on properties of phase locking). These artifacts are the result of nearby cortical surfaces originating from the same sensor-space sensors, which in turn creates a high sensor space signal overlap between the two nearby source space points. This is the point spread function of the inverse model that is location dependent.

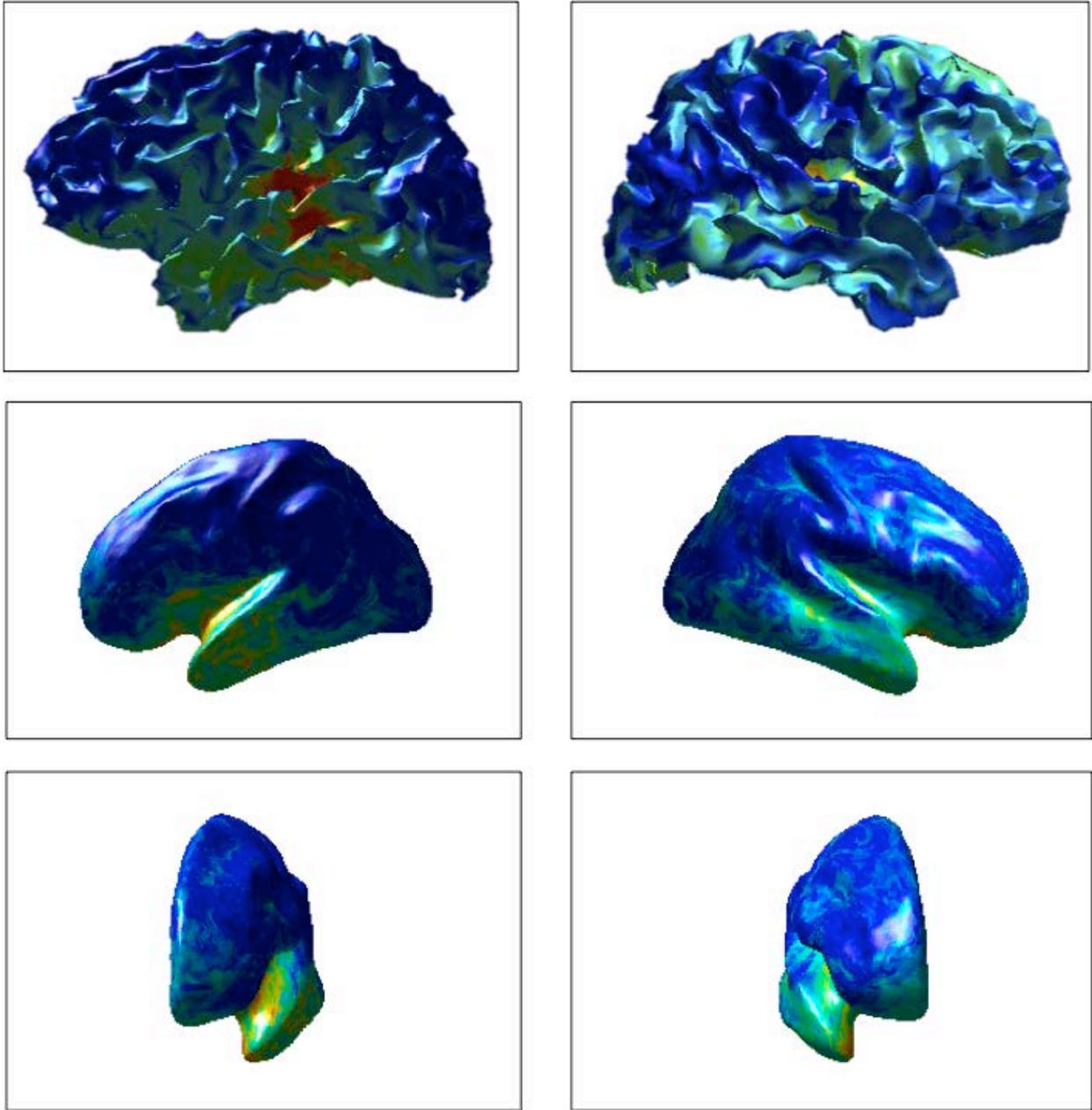
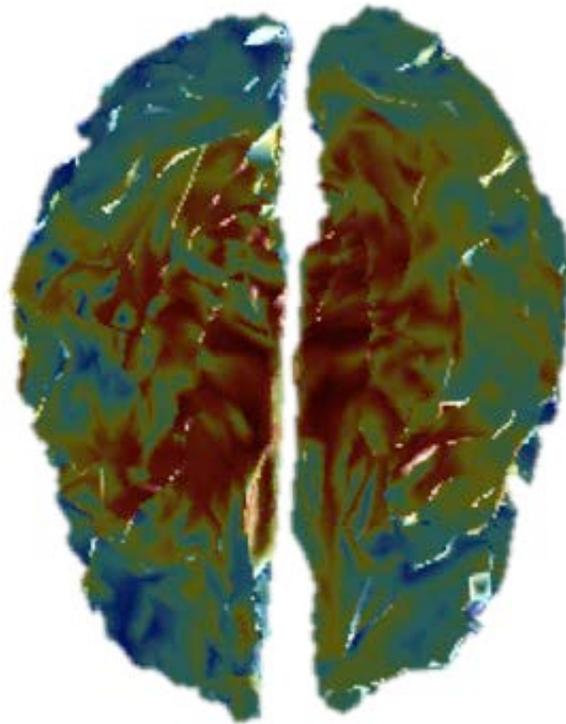


Figure 5-8 Empty Room Noise Example.

Left and right hemispheres of empty room eigenvector centrality are displayed in the left and right columns, respectively. That is, a phase locking graph between all vertices of empty room time series data were calculated and the eigenvector centrality of that graph calculated. Top is the display on the pial cortical surface while the second and third rows display the eigenvector centrality on the inflated surface.

At the base of the brain, where the SNR of MEG is poor due to the distance from the sensors, the inverse solution reconstructs those vertices with a high degree of overlapping contributions from sensors. As a result, the base of the brain has very strong phase locking connections with other vertices located at the base of the brain and weak connections to other cortical regions. The eigenvector centrality of these vertices also shows a high centrality score as the result of these strong connections to other strongly connected vertices at the base of the brain (Figure 5-9). Shown is the degree centrality of empty room data from the base of the brain. High centrality scores at the base are the result of large false positive clouds located at that region. Furthermore, the false positives are dominantly connected to other ventral locations (not shown).



**Figure 5-9 Visualization of empty room error at base of brain.**

This high signal contribution from poor SNR regions can result in spurious centrality scores. However, centrality scores of the empty room noise are not calculated in the subject phase locking analysis. The specific vertex-to-vertex connections are used in the permutation testing. Therefore, the empty room data is still a viable null graph for statistical network analysis. The results above are more important for understanding the inverse solution reconstruction.

## 5.4.2 Simulated Point Generator Phase Locking Properties

Simulation of point sources containing sinusoidal signals revealed the effects of non-random signals on the point spread function by examining the false positive cloud around the original signal generator. In Figure 5-10, we see the comparison of the seed location to every other vertex phase locking values in the absence (left) and presence (right) of a sinusoidal generator at the seed location. That is, a phase locking graph is calculated from the phase locking values between each point and every other point. To visualize the phase locking results of a single point, the seed, and every other vertex, we display the one-dimensional row (or column) vector in the adjacency matrix corresponding to the index of that seed point.

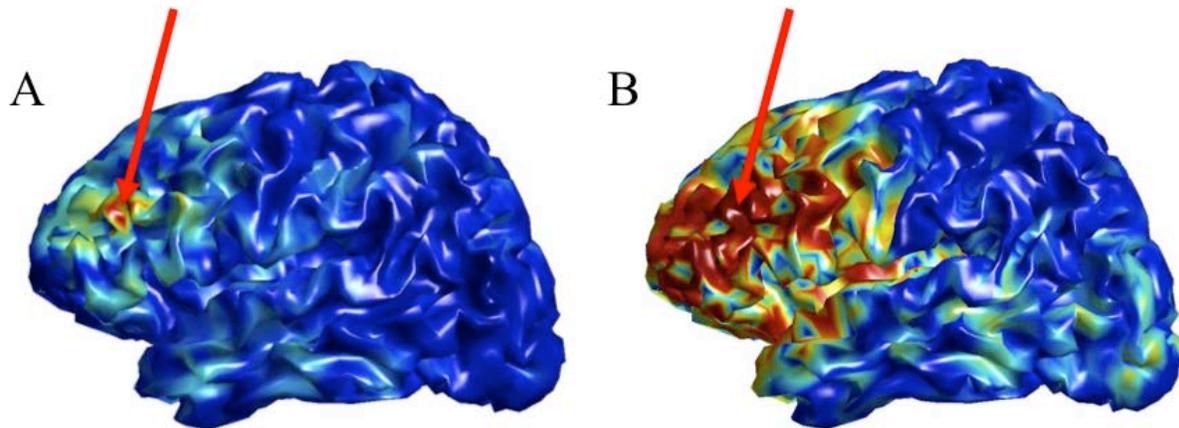


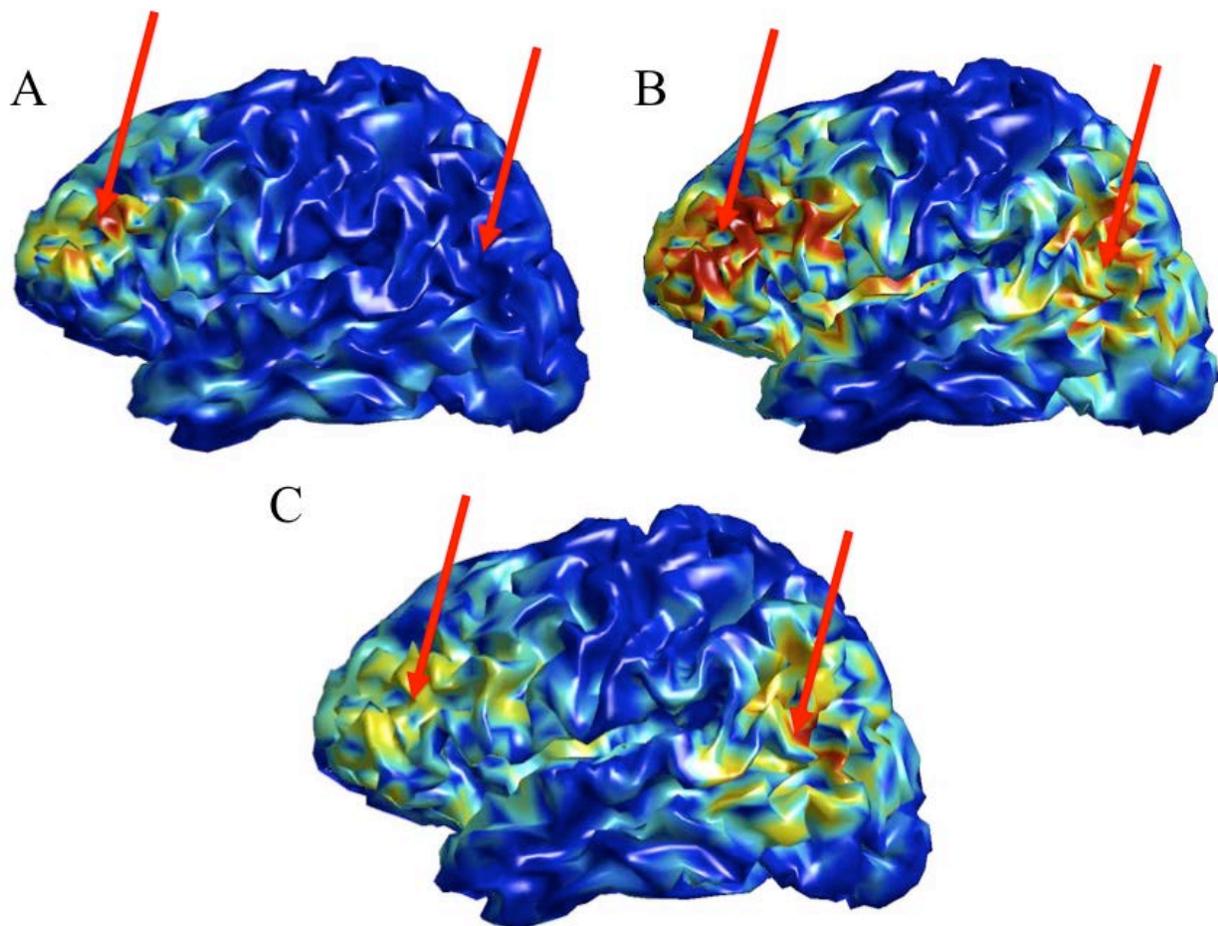
Figure 5-10 Effect of sinusoidal generator on point spread function.

Dipole sinusoidal generators increase the size of the point spread function in the presence of random noise only. The higher signal contribution is smoothed across the cortical surface as a result of the minimum norm calculation.

Next we simulated two point source sinusoids with a fixed frequency in which each point was located outside the point spread function of the other single generator. In Figure 5-11 (A) we have seeded the left arrow's dipole location and displayed that dipole's phase locking activity to every other vertex at an off-frequency to the generator. That is, the phase locking value for a frequency different than that of the simulated sinusoid to demonstrate the null result. Note that the opposite dipole that also contained a sinusoid at the same frequency (right arrow) shows little activity as a result of the off-frequency calculation. Calculating phase locking values at the same frequency as the sinusoids shows both regions displaying positive phase locking in Figure 5-11 (B). Similarly, if we seed the right arrow and view that dipole's phase locking activity to every other point we arrive at the image in Figure 5-11-C at the generator frequency.

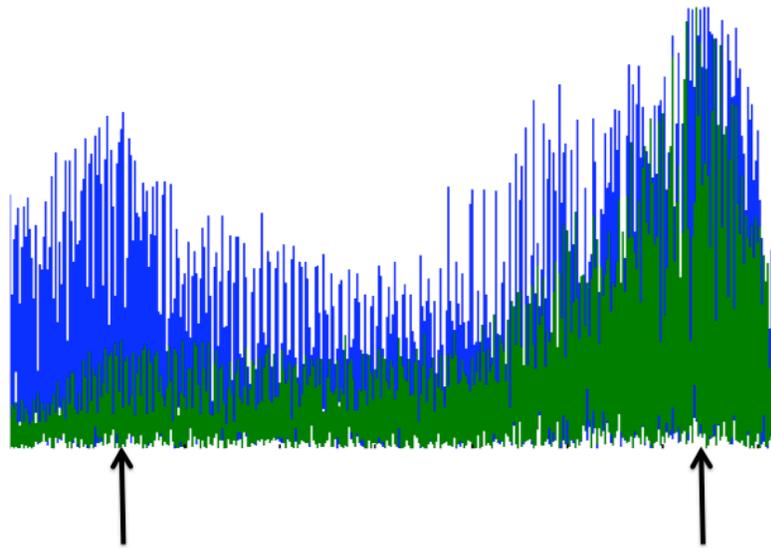
A 15 Hz sinusoid was generated as the dipole location pointed to by the red arrow. (A) is the phase locking value between the seed point and every other dipole location for a phase locking graph at 6 Hz. (B) is similar to (A) but with a phase locking graph calculated at a frequency of 15 Hz at which the sinusoid was generated in simulation. The larger point spread function is the result of spatial blurring of the high strength sinusoid onto nearby vertices resulting in a point spread function of false positive values.

Note that B and C are not equivalent. Phase locking is symmetric and the values between each of the seeded dipoles are equal, however, the point spread function is location dependent and therefore non-symmetric. In graph form, we guarantee that the adjacency matrix is mirrored across the diagonal, but to have symmetry in the forward and reverse directions of the point spread function would require rows to be symmetric.



**Figure 5-11 Two-point generators with overlapping phase locking.**

One can also visualize the phase locking value of a single seed region as a 1-dimensional plot of the phase locking value with respect to the vertex indexing rather than the cortical dipole locations plotted onto a surface. In Figure 5-12 we show the plot of a seed region (right arrow) index in the off-frequency (green) and at the sinusoidal frequency of the generator (blue) in which the second seed region (left arrow) is activated with a higher phase locking magnitude. As noted previously, nearby indices are only roughly close in cortical space. Given that restriction, we still note that qualitatively there is decay in phase locking values as a function of distance from the seed region in green from the right arrow.



**Figure 5-12 Point spread function of two point generators in one-dimensional vertex indices.**

In addition, we see the influence of non-symmetric row data in the phase locking graph as a high peak around the seeded point (right arrow) that extends to a phase locking value of one and decays. At the distal second seed region, the phase locking values do not peak to one but nevertheless decays from maximal intensity from the left seed region.

In (A) the result of seeding the frontal dipole location and using an off-frequency for estimation of phase locking. (B) two point generators at the sinusoid generator's frequency while seeding the frontal dipole location. In (B) we are able to identify the second generator's activity. Also note, that previous figures showed the point spread function without the second generator. The addition of a second source location reduced the point spread function. (C) is the same as (B) but seeding the posterior location. Note that there is asymmetry between (B) and (C).

The two arrows represent the indices of two sinusoids in a simulation. The plot shows the result of seeding the right most generators' index and the calculation of phase locking to every other index. Green is at an off-frequency with respect to the generator's sinusoidal frequency. Blue is at the same frequency as the sinusoid generator. We see at the second location, on the left, the blue plot shows a high phase locking value.

The interaction between three point generators was then investigated by adding a third point to the previous simulations mid-way between the two original dipole locations on the cortical surface. This point was also simulated to have the same frequency as the first two and thus we expect positive phase locking at all three regions at that frequency.

In Figure 5-13-A,B,C we show the point spread function of off-frequency phase locking when seeding each of the three points. At the correct frequency we then seeded the anterior most point (C) and calculated the one to all phase locking, which is displayed in Figure 5-13-D. Note that due to the decay of signal to noise as a function of distance, the centrally located seed region appears to have higher mean activity (larger magnitude and larger dispersion) while the most posterior region has a smaller mean activity. In (A), (B), and (C) the false positive cloud of three sinusoidal generators in a simulation. In (D) the aggregate of seeding the point located in (C) and the overlap in individual false positive clouds.

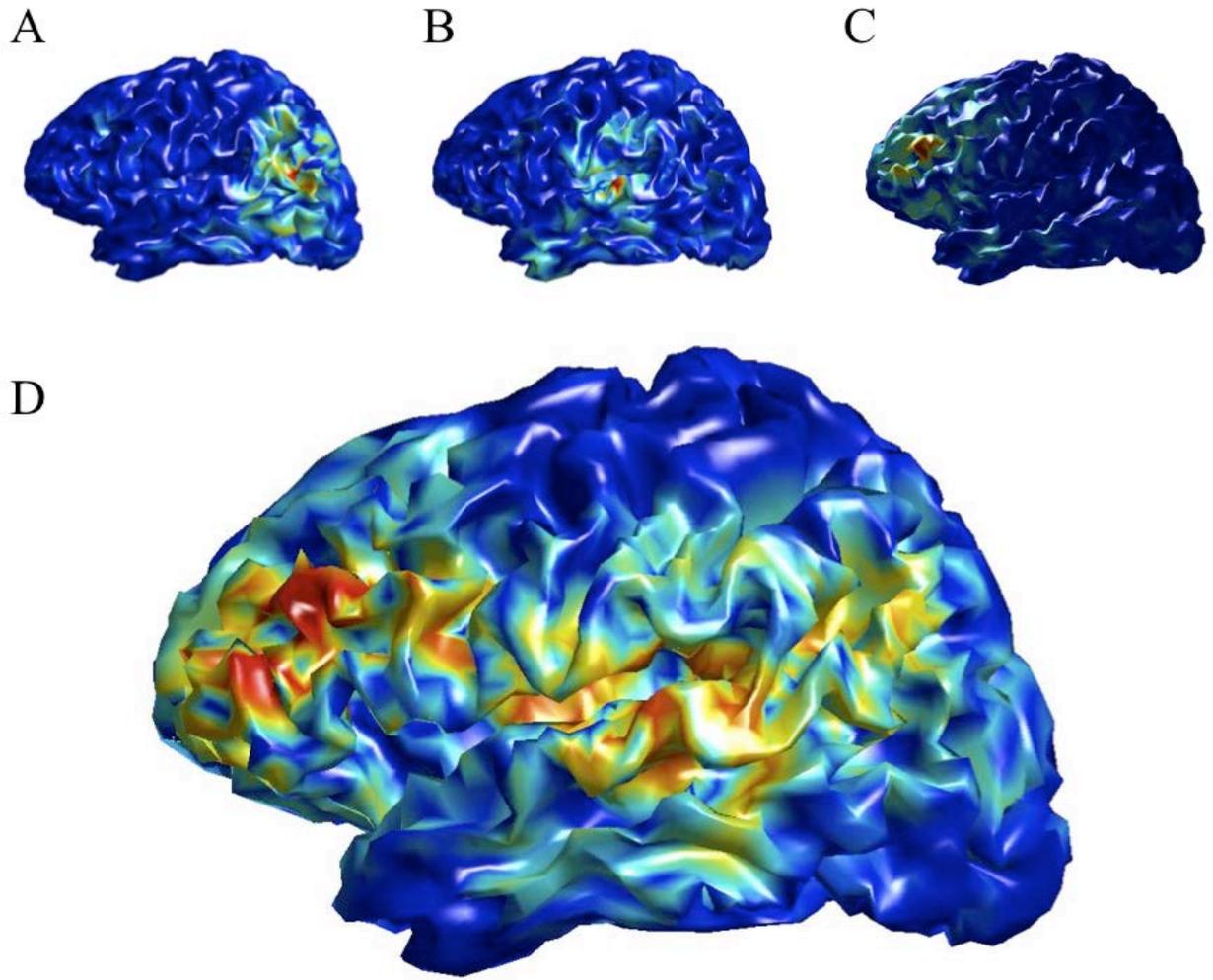


Figure 5-13 Three point simulation.

### 5.4.3 Phase Locking Resolution

Next we investigated the resolution of MEG phase locking by simulating two phase locked signals at incrementally closer distances. Two points, A and B were simulated on the cortical surface. Point B was fixed and 18 different simulations were conducted with different vertex locations for point A at incrementally closer cortical surface distances to B. Figure 5-14 shows the location of the seed points used in the simulation. Point B was fixed as the posterior most point. Vertices used in the simulation of two points at incrementally closer distances. Simulations fixed the posterior most vertex, B and moved the anterior point, A. 18 simulation distance pairs were used.

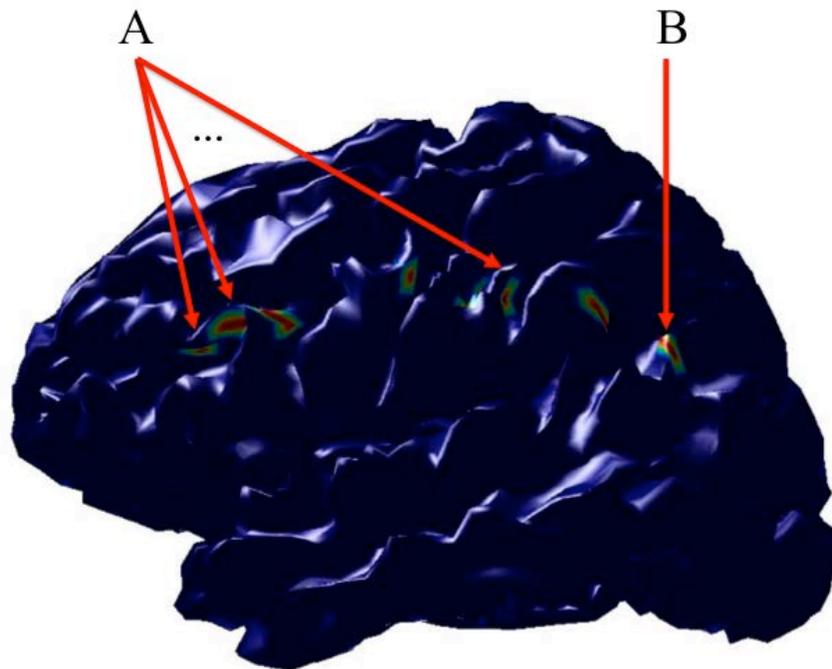


Figure 5-14 AB Simulation vertices.

Figure 5-15 shows the results of the A-B simulation at incrementally closer distances between A and B. At large distances (top and left), the point spread function for each point was non-overlapping. However, with decreasing distance between the seed points, the point spread function of the two points began overlapping. At extremely close distances, it is difficult to discern the differences between the two original point sources.

Note additionally in Figure 5-15 that if we compare the point spread of B when A is far away to the point spread function of B when A is extremely close we can see a distinct difference in its size and magnitude. Overlapping of nearby vertex generators tends to have an overall positive effect on the magnitude of the phase locking point spread function. This result is similar but opposite for anti-phase locked signals in that as they become closer they tend to cancel the point spread function around both points. This is a direct result of both the inverse model formulation that source space locations are a linear combination of multiple sensor space recordings in which case signals can constructively or destructively interfere as well as interference in magnetic fields.

Results are shown of two point simulations at incrementally closer points. The posterior most index is fixed while the anterior point is moved incrementally closer. Images are the result of seeding the anterior point and the calculation of that vertices phase locking with every other vertex. The distance between values decreases along rows. As the points converge, there is an overlap in the point spread functions of each seed region reducing the resolution.

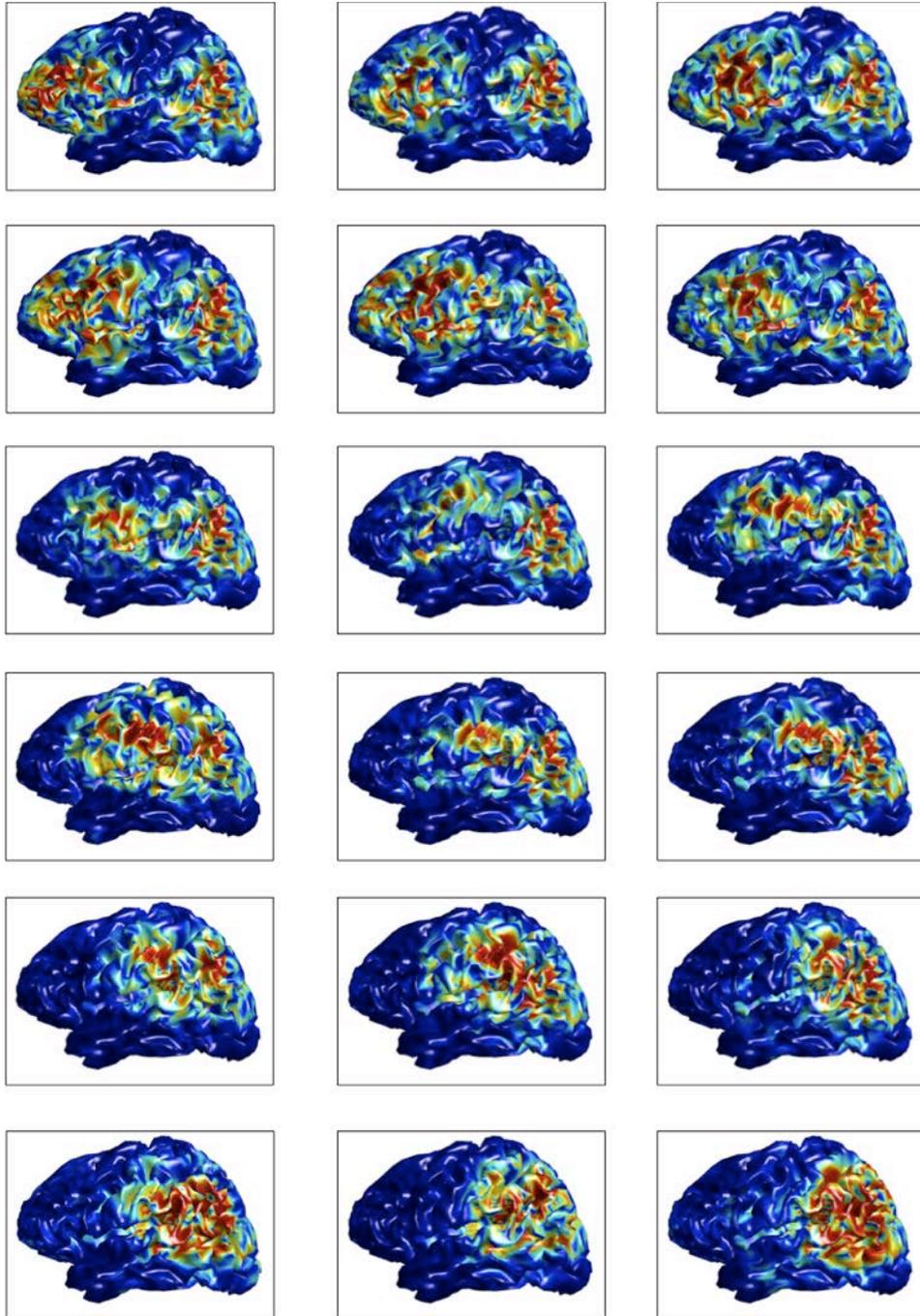


Figure 5-15 Point spread function overlap.

#### 5.4.4 Effects of Signal to Noise Ratio on Phase Locking I

In Figure 5-16 a low signal to noise ratio signal was applied at a seed region (red arrow) and the seed region to every vertex is displayed in A. Next, we increased the signal to noise ratio by increasing the signal amplitude relative to the simulation noise. The results, displayed on cortical indices in Figure 5-16-B show a false positive cloud extending over the entire brain. This implies that source space time series are roughly equivalent with similar combinations of the sensor space recordings at each dipole location. (A) is a very low amplitude and low SNR sinusoid at the vertex pointed to by the red arrow. (B) is an extremely high SNR sinusoid at the same vertex location. False positive phase locking is a function of the SNR of the underlying signal. Note however that only a single vertex sinusoid is unlikely in real MEG data.

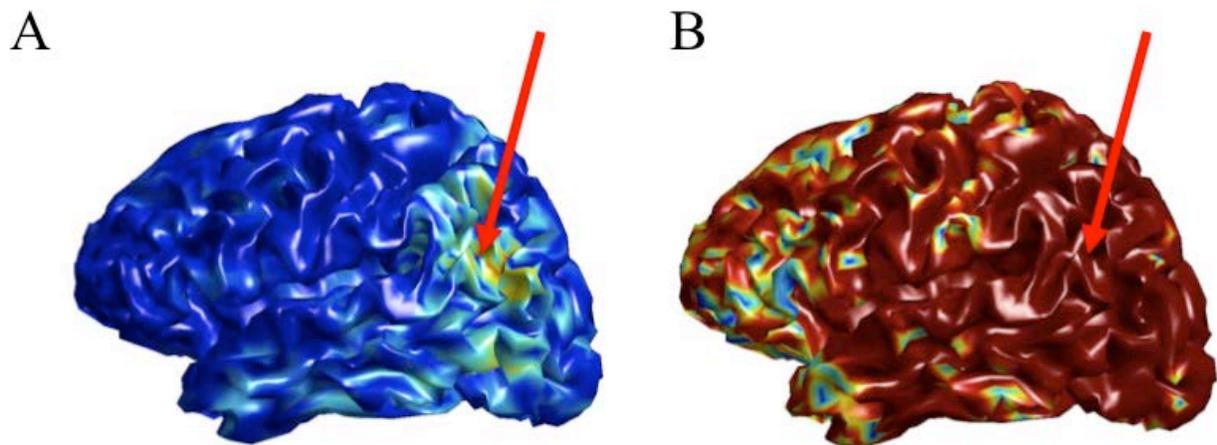
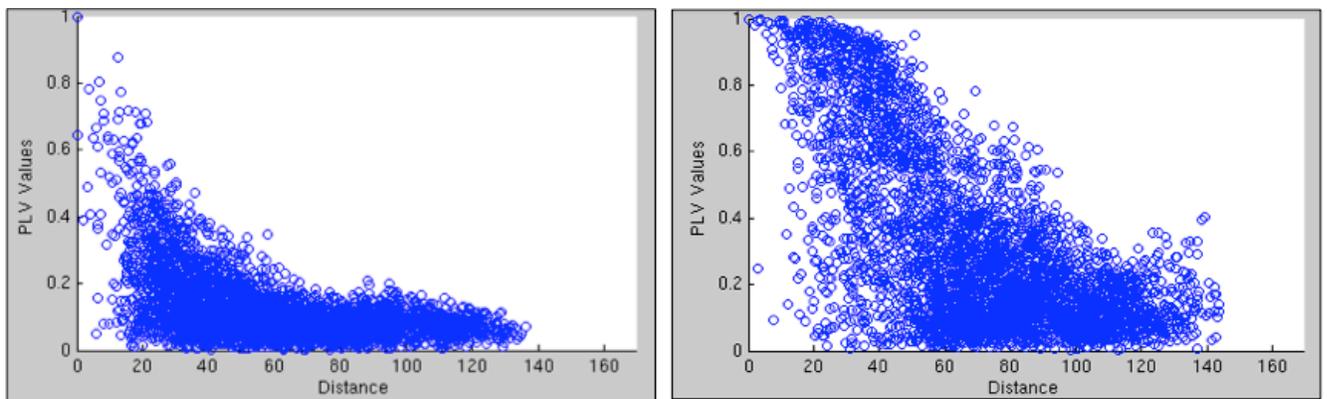


Figure 5-16 Source signal power and relation to false positive spatial correlations.

Two perfect sinusoids are phase locked across all frequencies. In addition, the minimum norm solution to the MEG inverse problem enforces spatially smooth outputs. In combination, this results in high magnitude frequency components having a larger false positive cloud. At low SNR, left, the phase locking value as a function of distance decays more rapidly than at a higher SNR, right.



**Figure 5-17 Phase locking false positives as function of distance.**

In Figure 5-17, we see the result of increasing the sinusoidal amplitude that creates a change in the phase locking distribution as a function of cortical distance. Increasing the SNR created a large false positive cloud within nearby dipole locations.

### 5.5.6 Positive Control

To determine the effectiveness of recovering true cortical phase locking the phase locking graph methods we applied to a positive control experiment in an effort to assess the recovery of *a priori* cortical activity. The positive control was calculated by simulating two sources on the cortical surface and attempting to recover the cluster by cluster statistical result using the methods from chapter 3. At both labeled regions in Figure 5-18, each vertex had the same sinusoid generated resulting in two phase locked signals between each vertex in both regions. An empty room dataset from the same coordinate vertices as the simulation data was used as the null dataset in the statistical graph calculations. Red regions are the vertices that were simulated to have phase locked sinusoids.

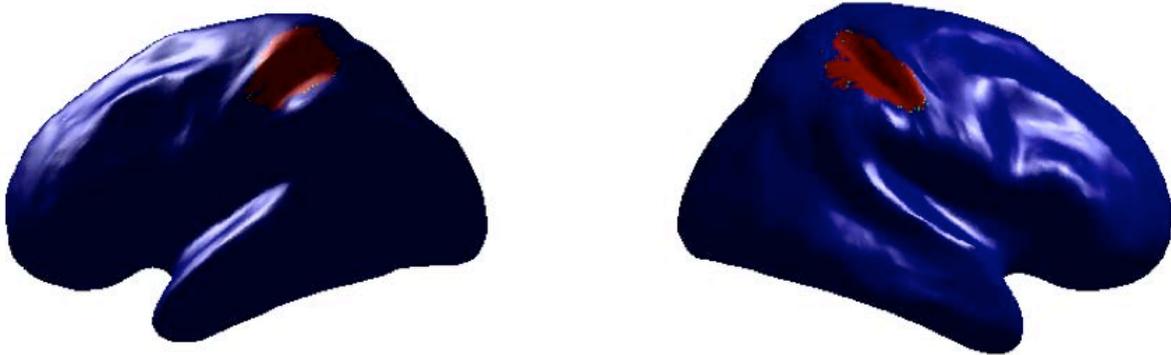
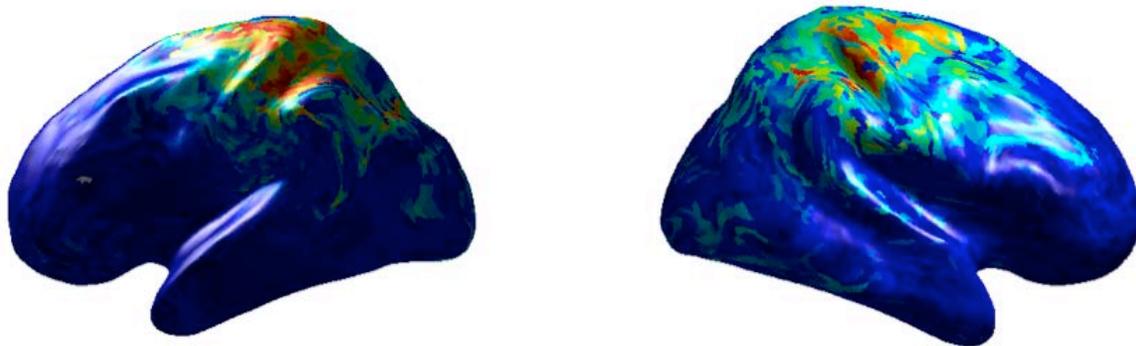


Figure 5-18 Positive Control Seed Regions.



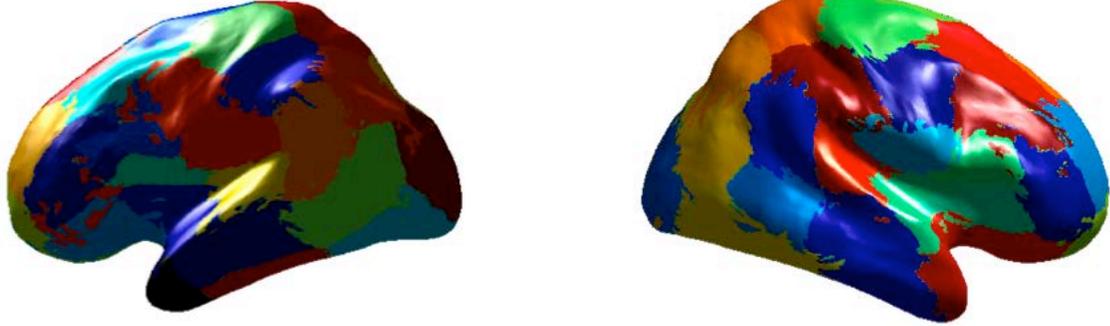
**Figure 5-19 Positive Control Results.**

Following simulation, phase locking values from a vertex from the left hemisphere seed region was displayed to every other vertex region. The dominant connection is a large positive region around both original simulation seeds (Figure 5-19).

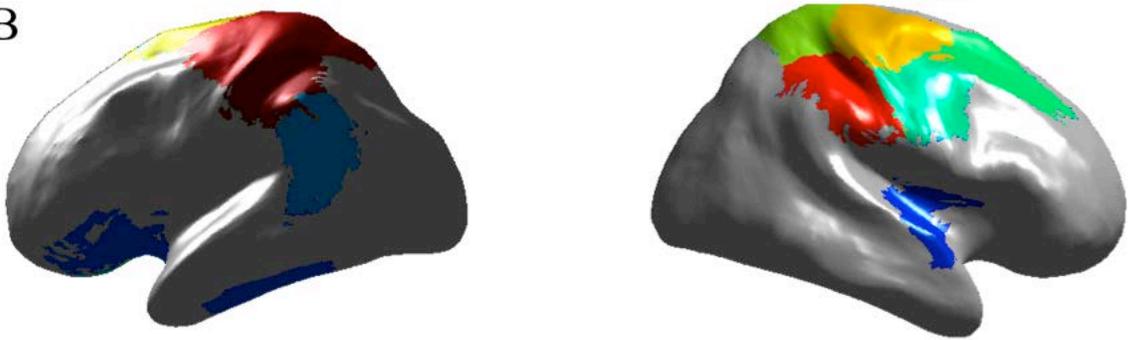
Figure 5-20 shows the clustering of the eigenvector centrality associated with this positive control simulation (Figure 5-20-A). In Figure 5-20-B, the statistically significant clusters are displayed after non-parametric testing of the cluster by cluster networks. For visualization, examples of nearby significant clusters are overlaid with the original seed location (in Figure 5-20-C, note that not all significant clusters are displayed).

Each cluster is smaller than the point spread function of the vertices used in the simulation thereby necessitating multiple clusters for coverage of the simulation regions. This is a result of the  $k$  value used in the  $k$ -means clustering. For optimal clustering of the entire input space of the eigenvector centralities, a large  $k$  must be chosen. As a result, this simulation resulted in cluster regions that overlapped with the original source location plus the expected size of the point spread function of that data. (A) the result of clustering the eigenvector centrality of the phase locking graph from the simulation with colors indicating cluster index. (B) is the degree centrality of those regions that were statistically significant following permutation testing. Red indicates the most network connections and blue indicating the fewest.

A



B



C

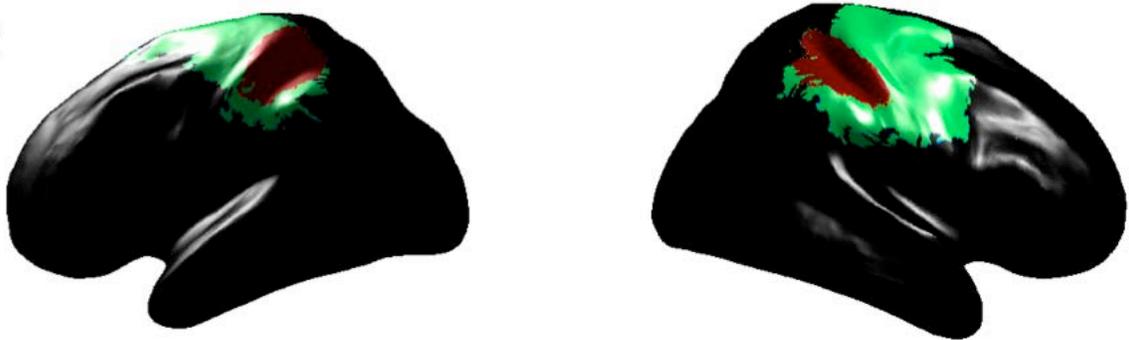
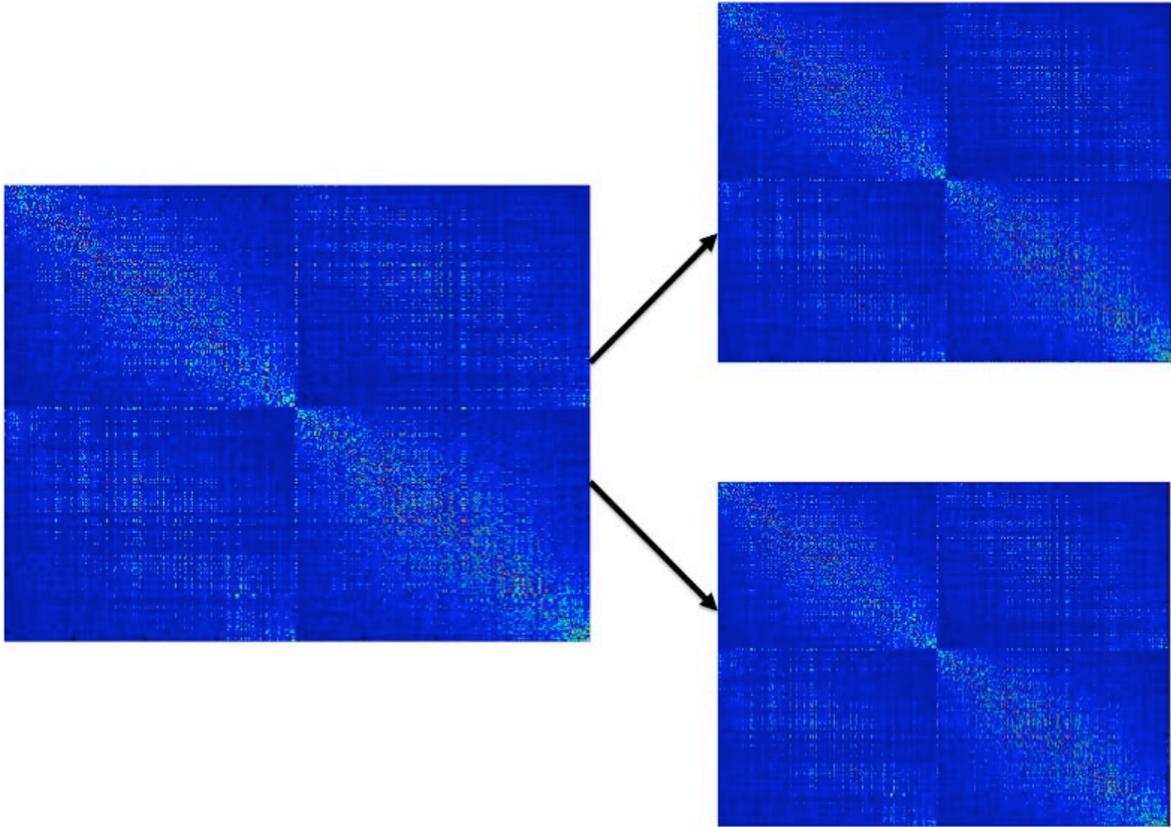


Figure 5-20 Significant Cluster Phase Locking Connectivity.

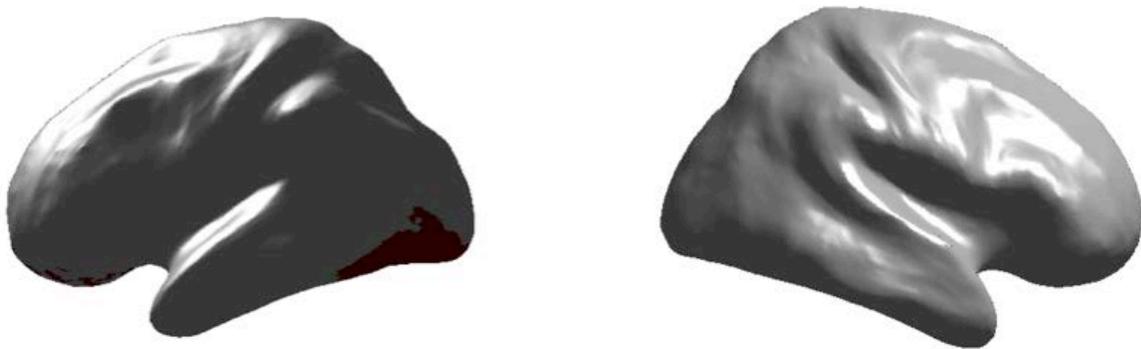
#### 5.4.6 Negative Control

Next, we tested that time series data in MEG without *a priori* phase locking signals will produce no network connections given the phase locking graph methods. A negative control dataset was created by splitting empty room time series, which we know *a priori* contain no neural signals, in sensor space into two sets consisting of the first half of the recorded time series and the second half (Figure 5-21). The two subsequent datasets were then used as distinct input “subjects” (n=2) and compared to a null dataset of a different empty room recording. The left image is the original phase locking graph of the full empty room dataset. The right is the first half, top right image, and second half, bottom right image, phase locking graphs after dividing the original empty room time series.



**Figure 5-21 Negative Control Data Creation.**

Figure 5-22 shows the result of a representative simulation of the statistically significant cluster networks found in the empty room dataset. Two small regions were identified as false positives at the base of the brain. In general, a few clusters were identified at the base of the brain during negative control experiments. This is most likely the result of poor signal to noise quality at that depth and a poor cortical surface reconstruction that results in spurious phase locking values. (Note that the significant clusters all exist at the base indicating that they are phase locked to other portions of the base of the brain, see above). Representative degree centrality of statistically significant clusters from the negative control studies. Regions at the base of the brain were statistically significant likely the result of large false-positive clouds resulting from poor SNR during cortical reconstruction.



**Figure 5-22 Negative Control Results.**

## 5.5 DISCUSSION AND CONCLUSION

Application of phase locking graph methods to MEG data has yielded results consistent with expectations. Statistical cluster based analysis of phase locking graphs produced the expected *a priori* null result for the negative control condition as well as the expected cortical region connectivity in the positive control. The use of empty room noise as a null graph for network analysis is specifically relevant for the correction of spatial correlations due to nature of the inverse models.

Future research should include an investigation into whole brain phase locking relations within the MEG sensor-space prior to cortical reconstruction. For non-linear models this is highly non-trivial and for a linear forward model it introduces some theoretical concepts that would require additional investigation. Namely, the interpretation of a linear combination of phase locking values on the higher dimensional cortical location. Additionally, phase locking or any functional connectivity estimation in sensor space is implicitly a relationship between pairs of sensor space-time series. This necessitates the inverse model being solved for relations between signals rather than the signals themselves.

It should be noted however, that the ill-posed construction of the inverse problem does not imply that solutions to source localization are mathematically incorrect. On the contrary, ill-posed problems typically have a non-singular and possibly infinite number of valid mathematical solutions. However, in practice only one state of the observed system gave rise to the measurement space. Therefore, any discrepancies in the mathematical solution with the real state of the system are not due to imperfections in the inverse solution – which is mathematically correct – but rather the nature of the formulation of the problem and our limited ability to observe the phenomena. The empty room null graph is therefore an estimation of the inverse solution not a comparison of the real neural activity and its observation.

Additionally, it is known that phase locking is not directly spectral amplitude dependent, however the phase locking value is dependent on the power spectrum magnitude at the frequency under investigation. Higher contributions of power at the given frequency give rise to a higher phase locking value from the combination of higher SNR and increased smoothing from the inverse solution. Given that the power spectrum of MEG data follows a  $1/f$  distribution, lower frequencies have higher spectral power that could bias estimates of phase locking.

MEG data is non-stationary, which affects phase locking measures. In general MEG data is WSS over approximately two minute time scales (see above). Wavelets, like all time-frequency measures, are less sensitive to non-WSS processes due to the temporal windowing. Within each window, changes in the joint distribution of the underlying stochastic process are limited to the window under investigation. Fourier methods in contrast integrate information over the entire time series. Non-wavelet time-frequency estimates would be similarly affected to wavelets.

While time-frequency methods can produce more accurate measures of changes to stochastic processes, the calculation of phase locking could still be affected. If a stochastic process changes

its underlying joint distribution that could result in changes to the instantaneous phase in subsequent temporal windows. If this occurs differently in the two time series under investigation, then a change in the overall distribution of phase differences will result. Only in the case in which a stationary shift occurs similarly with respect to phase in both time series will there be no net effect on the phase locking value.

## **6.0 APPLICATION TO WHOLE BRAIN PHASE LOCKING: EXPERIMENTAL DATA**

In this chapter, phase locking graph methodologies are applied to resting state MEG data of healthy human subjects. Significant networks are detected in intra- and inter- hemispheric connections and are interpreted within the context of previous literature regarding frequency-specific network communication in functional connectivity studies.

### **6.1 INTRODUCTION AND BACKGROUND**

Neuroimaging of spontaneous neural activity while a subject is at rest have been shown to exhibit patterns of functional connectivity similar to task driven experiments. Previous studies have used measurements of cerebral hemodynamics imaged via fMRI to demonstrate that correlations in activity can be used to identify functionally connected regions [66-71].

Resting state studies are believed to exhibit properties of the brain that are intrinsic to the brain's functioning without being driven by a task [72]. In addition, connections identified in task based studies are correlated when absent the task [73]. Electrophysiology methods for examining resting state functional connectivity have been studied in both EEG and MEG [74-78].

Identification of networks and their properties has already led to promising insights into the healthy brain's functioning. In addition, similar studies regarding functional connectivity have yielded information on the nature of diseased brain states. These intrinsic functional connectivity networks have already been used to examine Parkinson's disease [79-81], Alzheimer's disease [45, 82-84], and autism spectrum disorder [85, 86].

In addition, functional networks exhibit frequency-dependent communication in which functionally connected cortical regions are predominantly connected at specific frequency bands [34, 35, 87-89]. Phase locking is a spectral domain analysis method that explicitly analyzes signals at frequency bands allowing for investigation of these frequency-specific network communications.

Here we have applied methods for large-scale whole brain functional connectivity to the electrophysiological recordings of resting state MEG. This provides a mechanism for the analysis of whole brain networks at multiple physiological frequency bands.

## **6.2 METHODS**

### **6.2.1 Resting State Data Collection**

A 306-channel Elekta Neuromag scanner was used for recording of neuromagnetic signals. Head position coils were placed on the scalp to determine head position during recording.

Temporal spatial signal separation (tSSS) was employed to remove external magnetic field contributions in the signal recordings which can lead to spurious signals which appear to have originated from physiological changes but are in fact artifacts external to the MEG sensors [93]. Time series MEG recordings were preprocessed to remove excess noise and artifacts prior to connectivity analysis. The MEG signals were first bandpass filtered using cutoffs of 1 Hz and 50 Hz to remove the 60 Hz line noise and slow drifts. MEG scans were sampled at 1000 Hz and subsequently downsampled to 250 Hz.

### **6.2.2 Subjects**

Resting state (spontaneous) MEG signals were recorded from five healthy human subjects (3 males, ages 25-45 and 2 female, ages 20-25). All subjects were right-handed. Each subject participated in two 5 minute resting state MEG scans which were concatenated together in source space prior to pre-processing. In each scan the subject was instructed to fixate on a centrally placed cross-hair, eyes open while sitting upright. Each scan lasted five minutes. Prior to subject scanning an empty room dataset was collected which consisted of 10 minutes of MEG data collection without a subject present in the MEG. The institutional review board of the University of Pittsburgh approved all study procedures. Written consent was obtained from each participant prior to participation in this study.

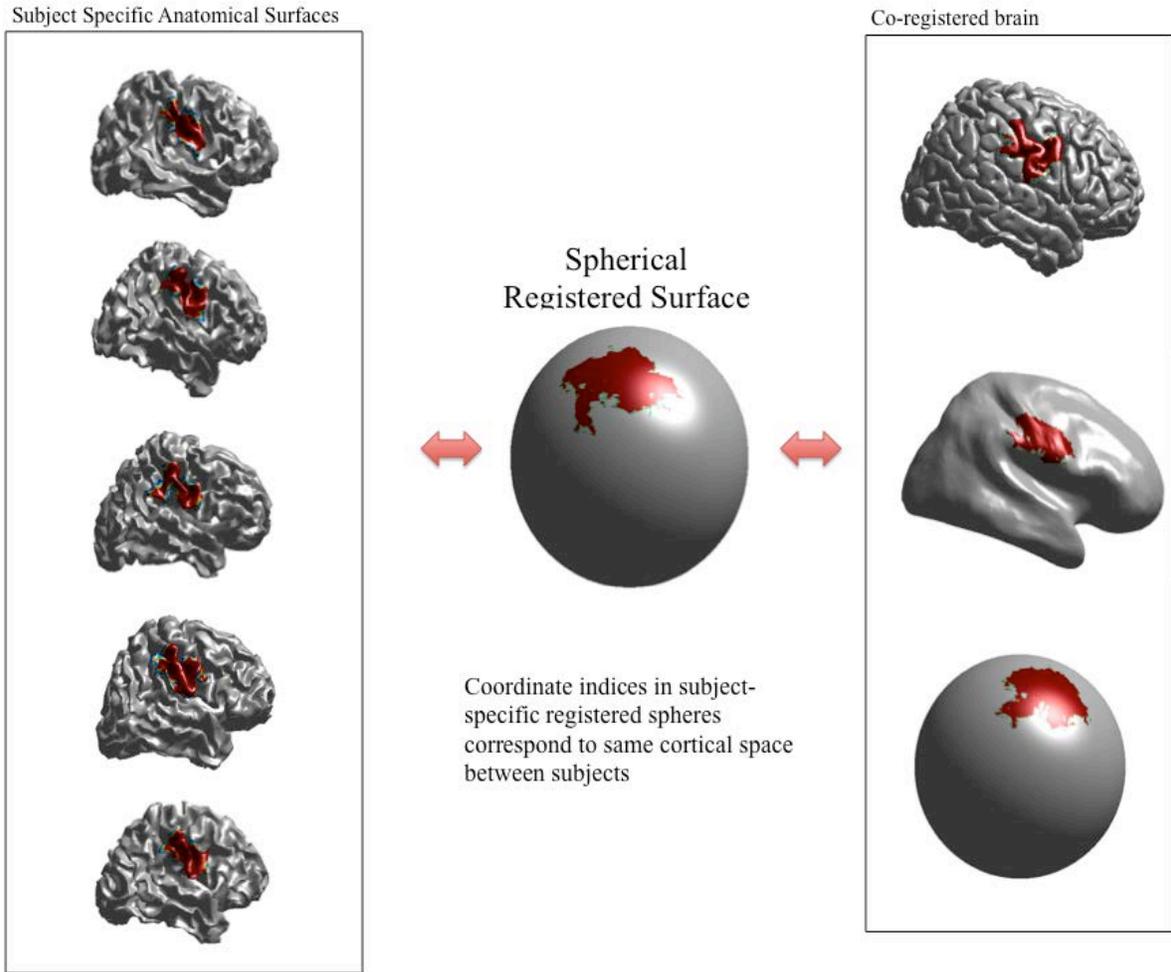
### 6.2.3 Subject registration

Each subject specific dataset was reconstructed onto the cortical surface, via methods established in section 5.3.1. To account for subject specific anatomical differences in cortical locations we performed a subject specific registration into a single global cortical surface space [90, 91]. This allows subsequent analysis to be performed in a single cortical space.

Inter-subject cortical registration was performed via the FreeSurfer registered spherical space. Each FreeSurfer cortical surface model is registered into a similar vertex space. However, different models can be exported according to the positioning of the vertex space relative to Cartesian coordinate systems without a loss of information. The pial surface representation represents the actual anatomical surface, however a spherical surface is also output which is the inflation of the single bounded anatomical surface of each hemisphere into a spherical shape. The spherical vertices were used in the clustering procedure as described in section 3.3.2, which ensured spatially smooth clusters.

A spherical coordinate system is also output for each subject from FreeSurfer [92-95]. This registered spherical system ensures that the same coordinate on the registered sphere from subject A and the subsequent coordinate on the registered sphere of subject B correspond to the same physiological coordinates in each subject-specific pial surface. For instance, the motor cortex for subject A in the pial surface and the motor cortex for subject B, when mapped onto the registered spherical coordinates will be nearly equivalent in the registered cortical sphere. In this way, we can create coregistration between subject anatomies by interpolating between subject specific anatomies.

We used the Colin atlas model as the global surface. Each subject specific anatomical cortical location was inflated into the registered spherical coordinates. Due to the downsampling of the cortical mesh during the application of the inverse model, it is necessary to interpolate the downsampled mesh onto the original registered spherical coordinates of that subject. A transform is then calculated between the subject specific spherical coordinates and the Colin atlas. This method allows inter-subject methods to be applied in the Colin space on all coregistered subject data (Figure 6-1). A similar inverse transform can be applied to map global metrics onto subject specific vertices.



**Figure 6-1 Cortical surface registration to a common atlas.**

In the center is a cluster displayed on a registered spherical surface. Each subject has a transformation between the registered spherical coordinates and the subject-specific anatomy of the cortical surface. In the left column, each subject-specific anatomical cortical surface displays the same region as center and right. In the right column is the registered surface of the Colin atlas showing the same region in the pial, inflated and spherical coordinates of Colin.

## 6.2.4 Phase Locking Analysis

We calculated the phase locking in neurophysiologically relevant bands by averaging over integer frequency values within a band. Each integer frequency phase locking graph was calculated for each subject specific cortical surface. The phase locking graph adjacency matrices were then averaged across the integers composing the frequency band. Averaging nearby frequencies produces Gaussian error terms due to the shape of the wavelet envelope. The bands used for this analysis correspond to: theta – 4-8 Hz, alpha – 8-13 Hz,  $\beta_{low}$  – 13-20 Hz,  $\beta_{high}$  – 20-30 Hz. Phase locking graphs at integer frequencies between 4 and 30 are averaged according to physiologically relevant frequency bands (Figure 6-2).

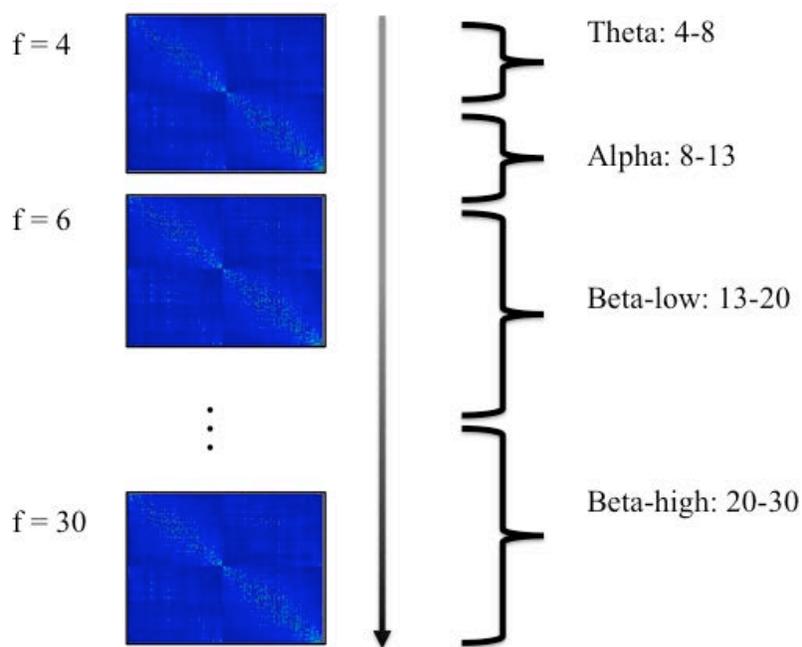
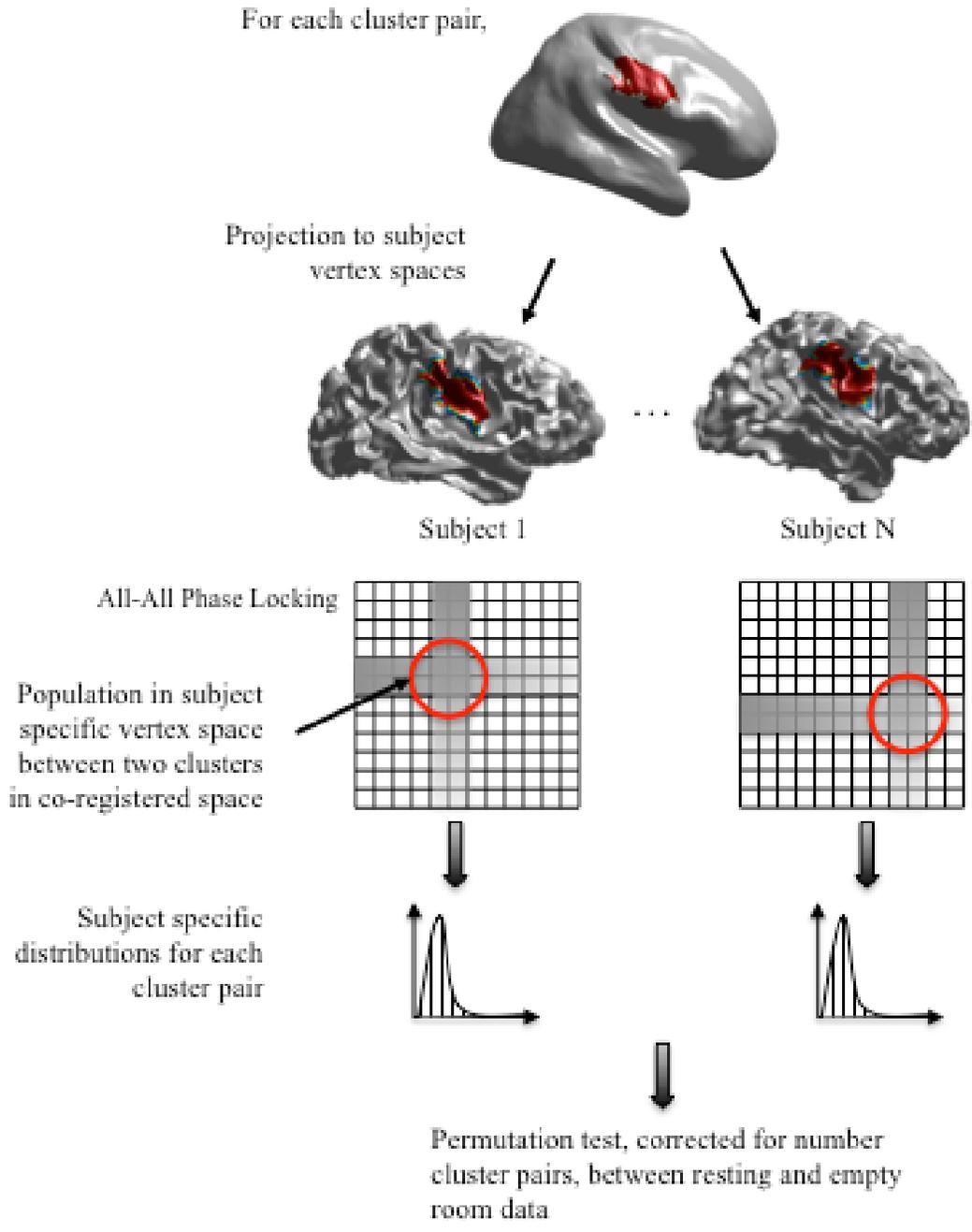


Figure 6-2 Averaging of Phase Locking Bands.

Analysis of healthy human subject resting state MEG proceeded according to the methods from chapter 2 for the analysis of whole brain functional connectivity graphs. Briefly, phase locking graphs were generated for each subject in each band and a similar procedure was used for the analysis of empty room data as the null hypothesis graph. Eigenvector centralities were computed for each phase locking graph in subject level anatomies. Those centralities were then transformed into the registered cortical space of Colin and averaged to obtain the global centrality measure at that band. Statistical analysis of cluster level pairs were then calculated according to the methods from section 3.4 via a non-parametric permutation test (Figure 6-3).



**Figure 6-3 Phase Locking Permutation Testing Between Subjects.**

Each cluster, top, in the global cortical surface space is transformed onto the subject specific vertices that correspond to that cluster. The process is repeated for a second cluster from which the cluster to cluster network analysis is to be performed. The two cluster vertices' intersection is found in each subject specific phase locking graph and a distribution for each subject is obtained. A similar procedure is used on empty room datasets and the process is repeated for each frequency. These distributions are used as input to the permutation testing.

Due to both the empty room spatial correlation as well as increased noise at more distant cortical locations the distributions between inter-hemisphere cluster populations and each of the intra-hemisphere populations were different enough to warrant individualized statistical treatment. Therefore, we calculated the critical values for within each hemisphere and between the hemispheres separately using individual permutation testing in order to derive inter-hemispheric and intra-hemispheric critical values. This was accomplished via three Monte Carlo analyses of the permutation tests to characterize the underlying distribution of phase locking in each intra-hemispheric distribution and the inter-hemispheric distribution. Without this correction, a single connection type dominated the significant values due to a higher mean.

### **6.3 RESULTS**

Centrality scores associated with subject and group level statistics are presented for each frequency band of interest. Following statistical testing, significant cluster level networks were found at the group level in each frequency band. Networks were separated for graph visualization purposes into inter-hemisphere and intra-hemisphere results.

### 6.3.1 Centrality Measures

A phase locking graph was computed for each subject at each frequency band. Eigenvector centralities are displayed on subject specific anatomies in Figure 6-4. The left and right hemispheres are shown. The color indicates the relative importance of that vertex within the original phase locking graph with red indicating the most important and blue indicating the least important. Within each frequency bands box the subject ordering is the same. So to compare the centrality measures at each frequency for subject one, it is the first row in each box.

Some subjects have a high degree of overlap between adjacent frequencies, for instance subject two in beta-low and beta-high have frontal cortex dominance in the left hemisphere and parietal dominance in the right hemisphere across both frequencies. This right hemisphere activity is also present in the alpha band for subject two, however the left hemisphere of alpha is not localized to the same region. Additionally, in the theta band the right hemisphere activity shifts anteriorly while the left-hemisphere centrality is not as concentrated.

In general, subjects show overlap between beta-low and beta-high. While the alpha band and theta band tend to have distinct localizations. Notable exceptions are the heavy dorsal activity corresponding to somatosensory areas in the beta-low frequency band for subject four and the distinct absence in the beta-high frequency for the same subject.

In the global space, the average of subject specific centrality scores was computed in the registered Colin cortical space. In Figure 6-5, we see the result of averaging of the theta network centralities. In general, this averaging procedure tended to create focal regions of high intensity centrality scores. In theta band, we see the left frontal region, and left visual areas as well as the right premotor areas with high centrality. Additionally, there is a lack of temporal and right lateral frontal cortex centrality.

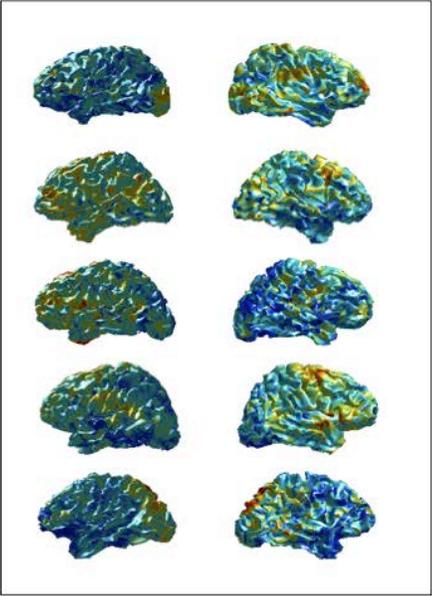
In contrast to the theta network that had large dispersed areas of high centrality nodes, the globally averaged alpha network in Figure 6-6, shows an extremely large centrality score in the parietal cortex with additional contributions moving anteriorly. In addition, the anterior most locations of the frontal cortex in both hemispheres also have a high centrality.

The beta-low centrality visualization in Figure 6-7, shows a similar pattern in the right hemisphere to that of the alpha network with a large parietal and premotor centrality areas but of a lower magnitude than the alpha band. The anterior portions of the dorsal activation in the left hemisphere are also present but again at lower amplitude than in the alpha band. In addition, the frontal regions are still activated symmetrically in beta-low network as they were in the alpha network.

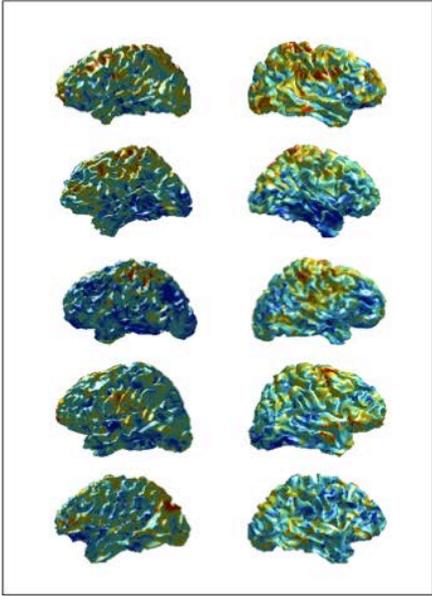
The beta-high centrality scores, in Figure 6-8, are more concentrated within the right and left parietal areas than beta-low. In addition, high centrality values are present in the somatosensory cortex as well as the frontal cortex.

In Figure 6-9, the mean across all centrality scores in each frequency band are displayed as well as the standard deviation. Dorsal areas of both hemispheres have high centrality across all frequencies as well as the left hemisphere frontal and occipital lobes. The right hemisphere also has a highly distributed variance across the cortex indicating high variability across the frequency bands. Subject specific,  $n=5$ , eigenvector centralities of four physiological frequency bands are displayed. Values are representative of the relative importance within the graph. Red values are the most important and blue values are least important within the graph. Subjects ordering are the same in each frequency band.

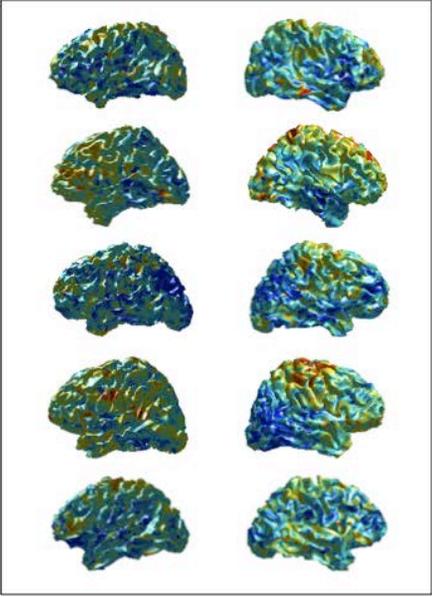
Theta



Alpha



Beta Low



Beta High

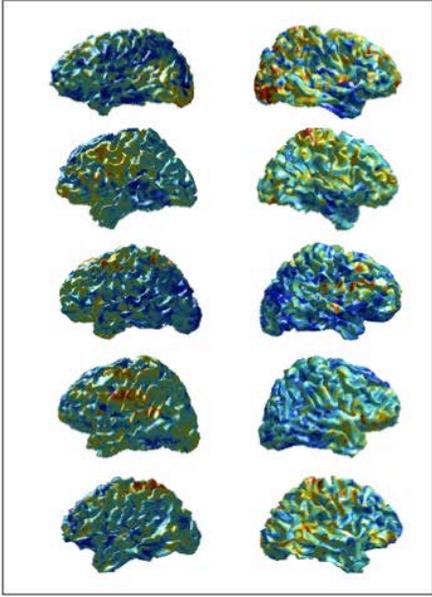
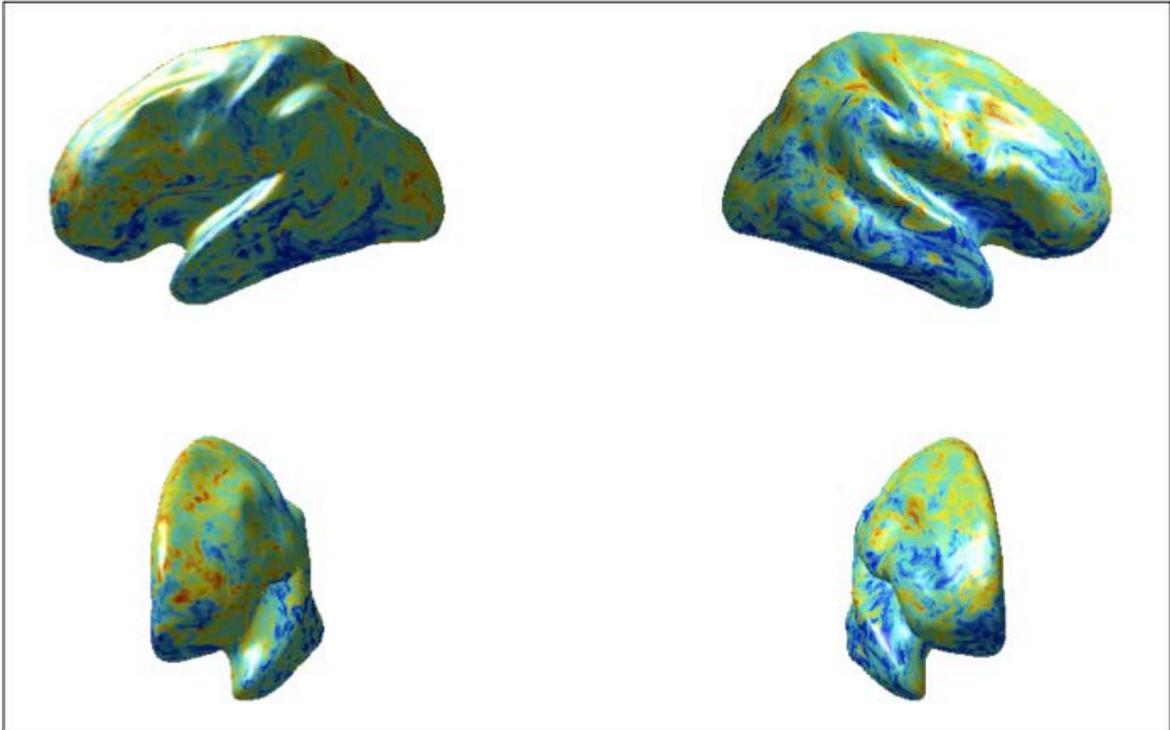
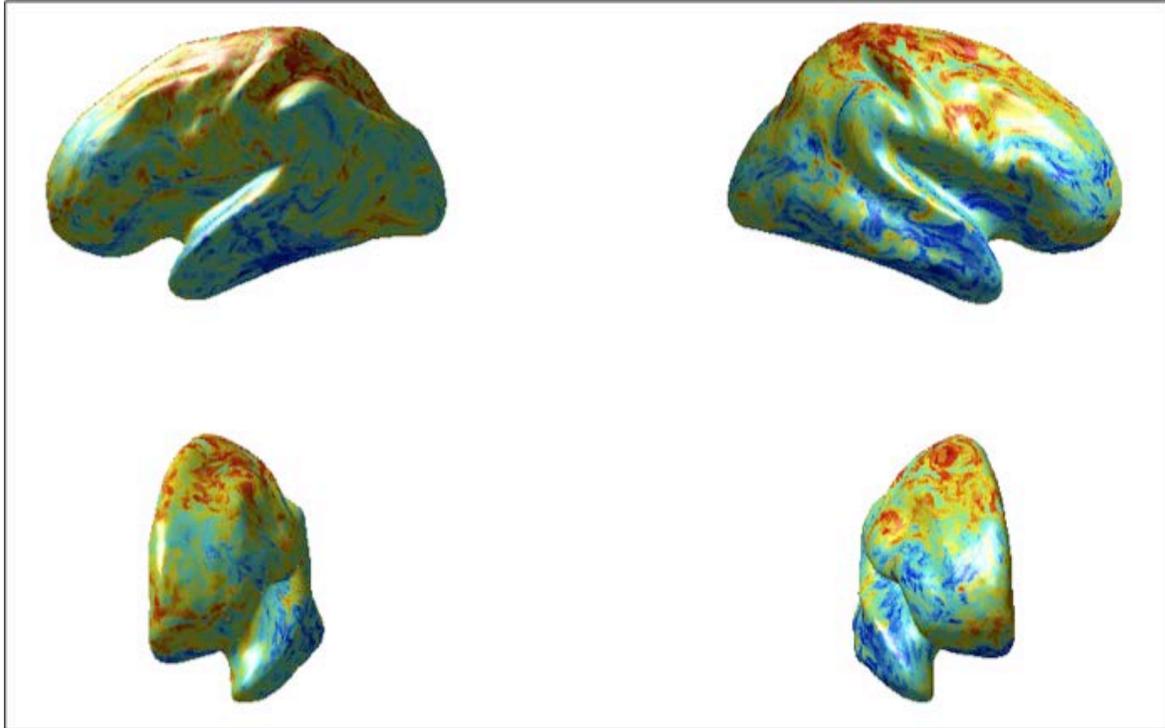


Figure 6-4 Subject Specific Eigenvector Centrality of Resting State Phase Locking.

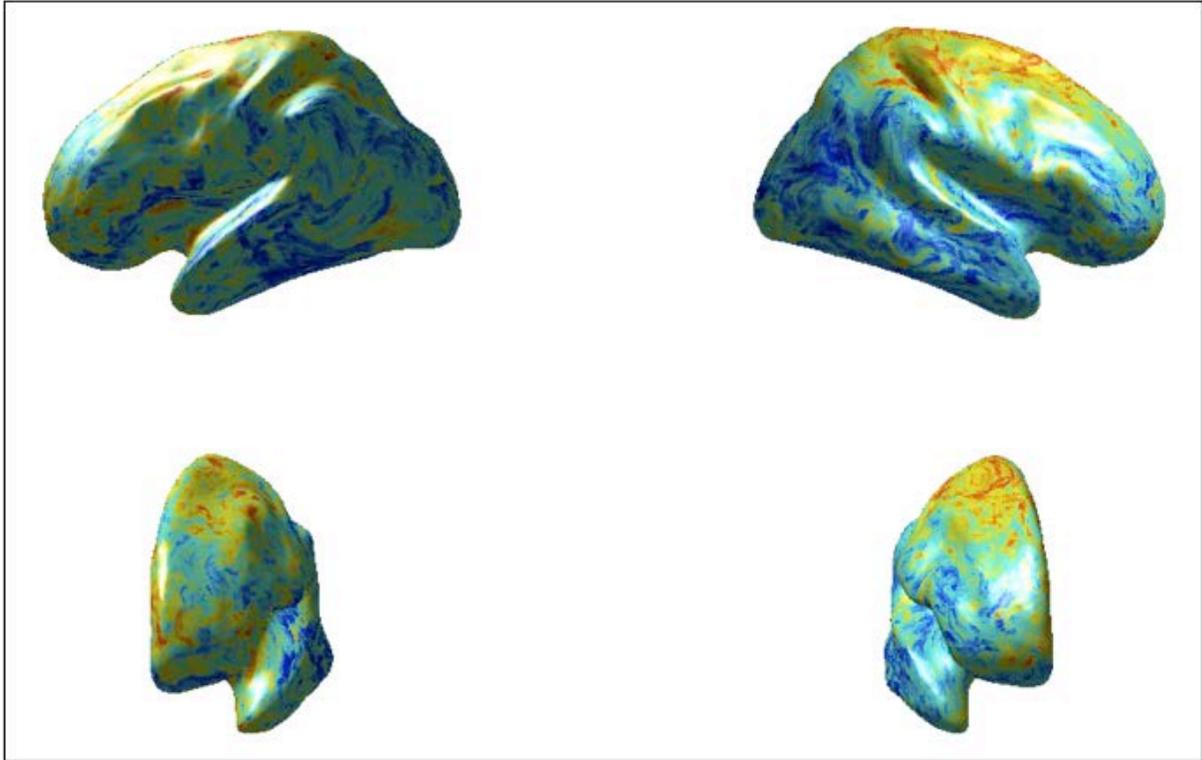


**Figure 6-5 Theta Band Network of EVC.**



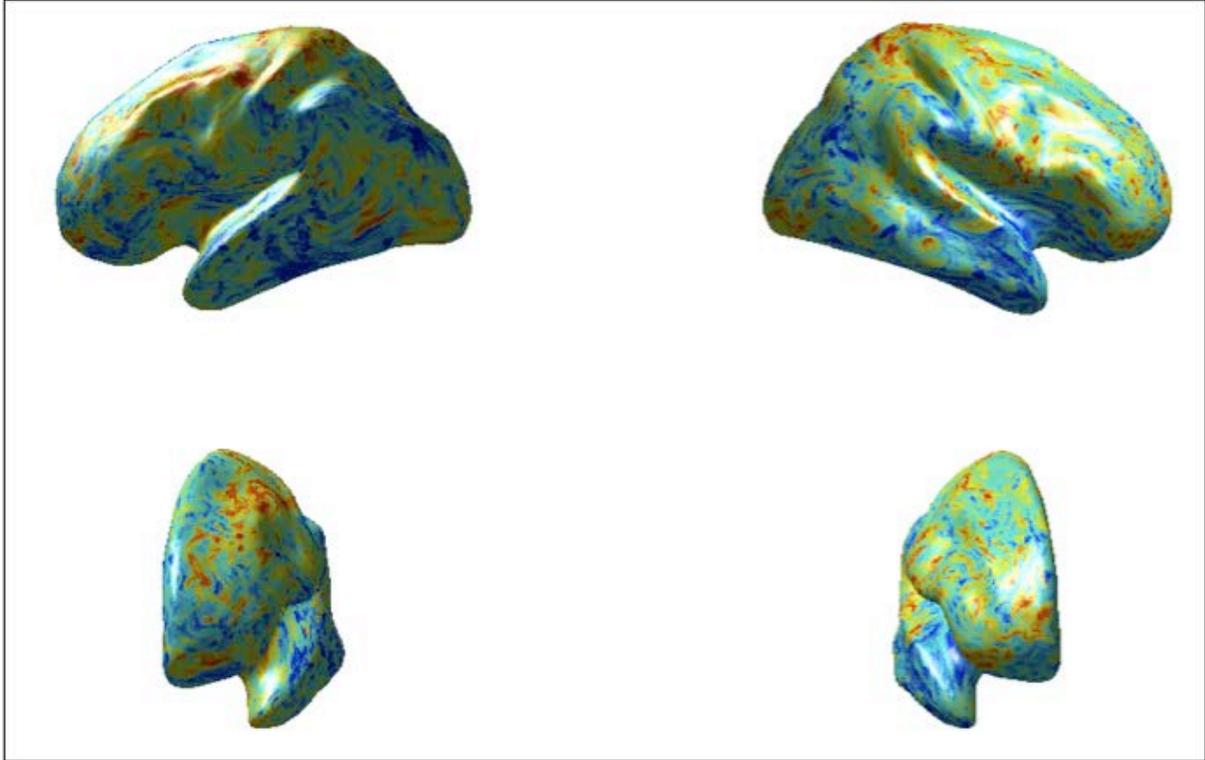
**Figure 6-6 Alpha Network of EVC.**

In Figure 6-6, color indicates the relative importance of the node within the network at the given frequency band. Red indicates highly important nodes while blue indicates a low importance node.



**Figure 6-7 Beta-Low Network of EVC.**

In Figure 6-7, color indicates the relative importance of the node within the network at the given frequency band. Red indicates highly important nodes while blue indicates a low importance node.



**Figure 6-8 Beta-High Network of EVC.**

In Figure 6-8, color indicates the relative importance of the node within the network at the given frequency band. Red indicates highly important nodes while blue indicates a low importance node.

## Mean

---



## Standard Deviation

---



**Figure 6-9 Mean and Standard Deviation of EVC Across All Frequencies.**

In Figure 6-9, color for the mean EVC indicates the relative importance of the node within the network at the given frequency band. Red indicates highly important nodes while blue indicates a low importance node. In the bottom standard deviation image, red indicates areas of high variability between frequencies while blue indicates areas of low variability.

### 6.3.2 Theta Band

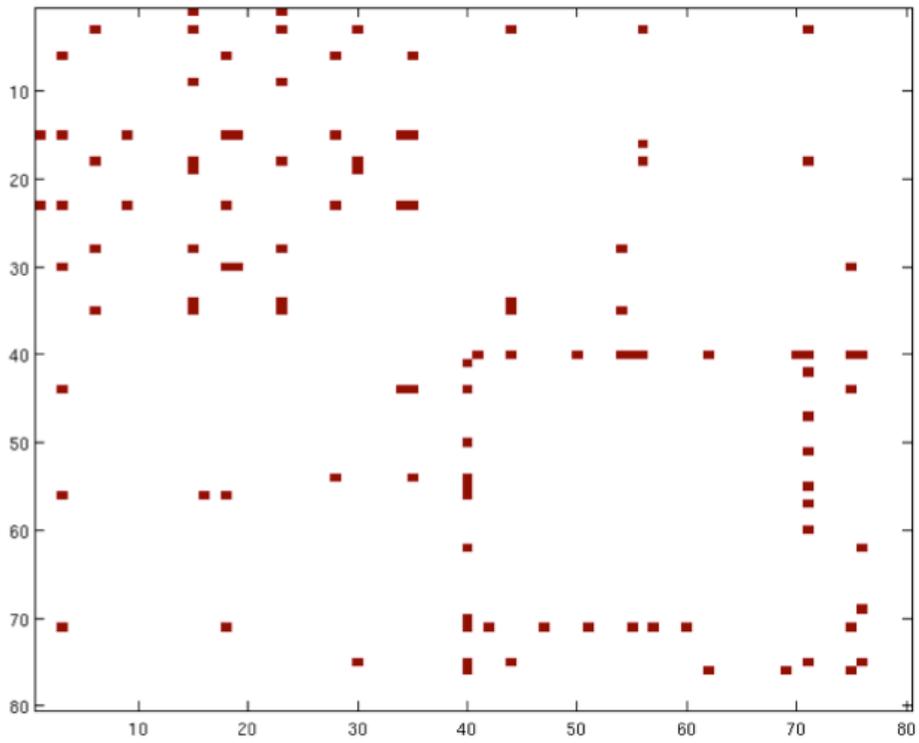
Theta band activity was calculated in the frequency range of 5-8 Hz. The adjacency matrix of significant cluster interactions is given in Figure 6-10 for the theta band. Significant cluster to cluster networks are present as red dots in the adjacency matrix entries. The matrix is symmetric along the diagonal as a result of symmetry in phase locking calculations. The first 40 clusters are in the left hemisphere while the second group of 40 clusters is in the right hemisphere. As a result, the upper-left quadrant corresponds to intra-hemisphere activity in the left hemisphere while the lower right quadrant corresponds to intra-hemisphere activity within the right hemisphere. The lower left (and upper right) quadrants correspond to the inter-hemisphere clusters.

In Figure 6-11, the intra hemisphere cluster level networks are shown. The top and middle images are the degree centrality of the clusters. The color of each cluster corresponds to the number of significant connections incident on that cluster. Red indicates a large number of connections while blue represents a small number of cluster connections. Grey areas are locations in which no significant networks were found.

In the bottom of Figure 6-11, we have overlaid a graph onto the degree centrality image of the brain. Red spheres correspond to cluster centroids and the size of the centroid is proportional to the number of connections incident upon that cluster. Black lines are edges representing significant connections between clusters.

Significant clusters in the theta network are centered around the frontal cortex and moving dorsally towards posterior areas terminating before parietal cortex. Networks in theta tend to connect frontal areas with occipital cortex as well as temporal cortex. Areas in the dorsal region of frontal cortex are also connected with visual cortex. The single network area in the right parietal cortex is connected with the frontal region.

Figure 6-12 shows the degree centrality of the inter-hemisphere connections of the theta band network. The left anterior temporal region in the left hemisphere is heavily connected to many regions in the right hemisphere. In addition, the frontal cortices of both hemispheres are highly connected. 40 clusters per hemisphere were used. The 80 cluster pairs are shown in the adjacency matrix. The first 40 correspond to clusters located in the left hemisphere while the last 40 correspond to clusters located in the right hemisphere.



**Figure 6-10 Theta Band Adjacency Matrix.**

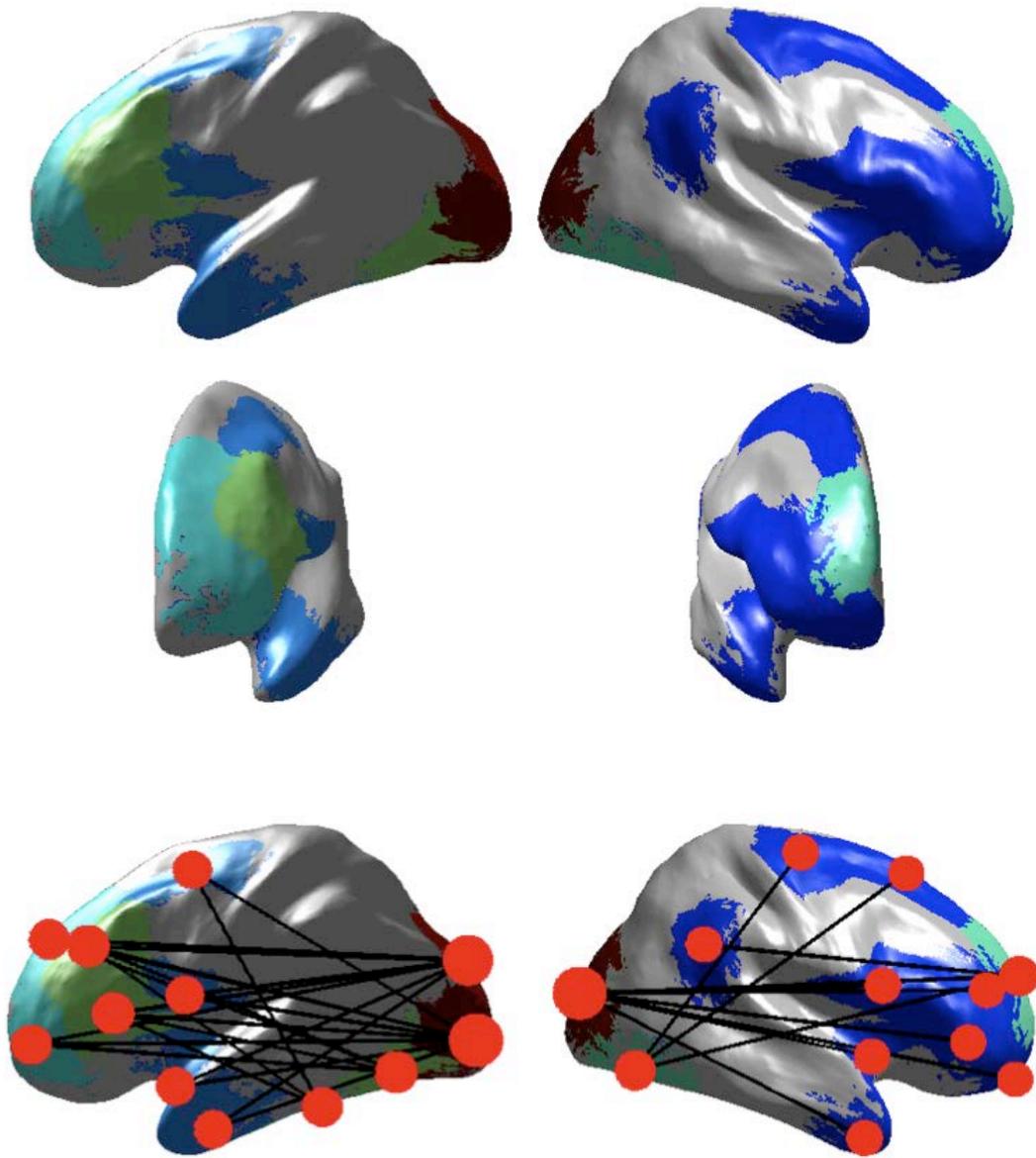
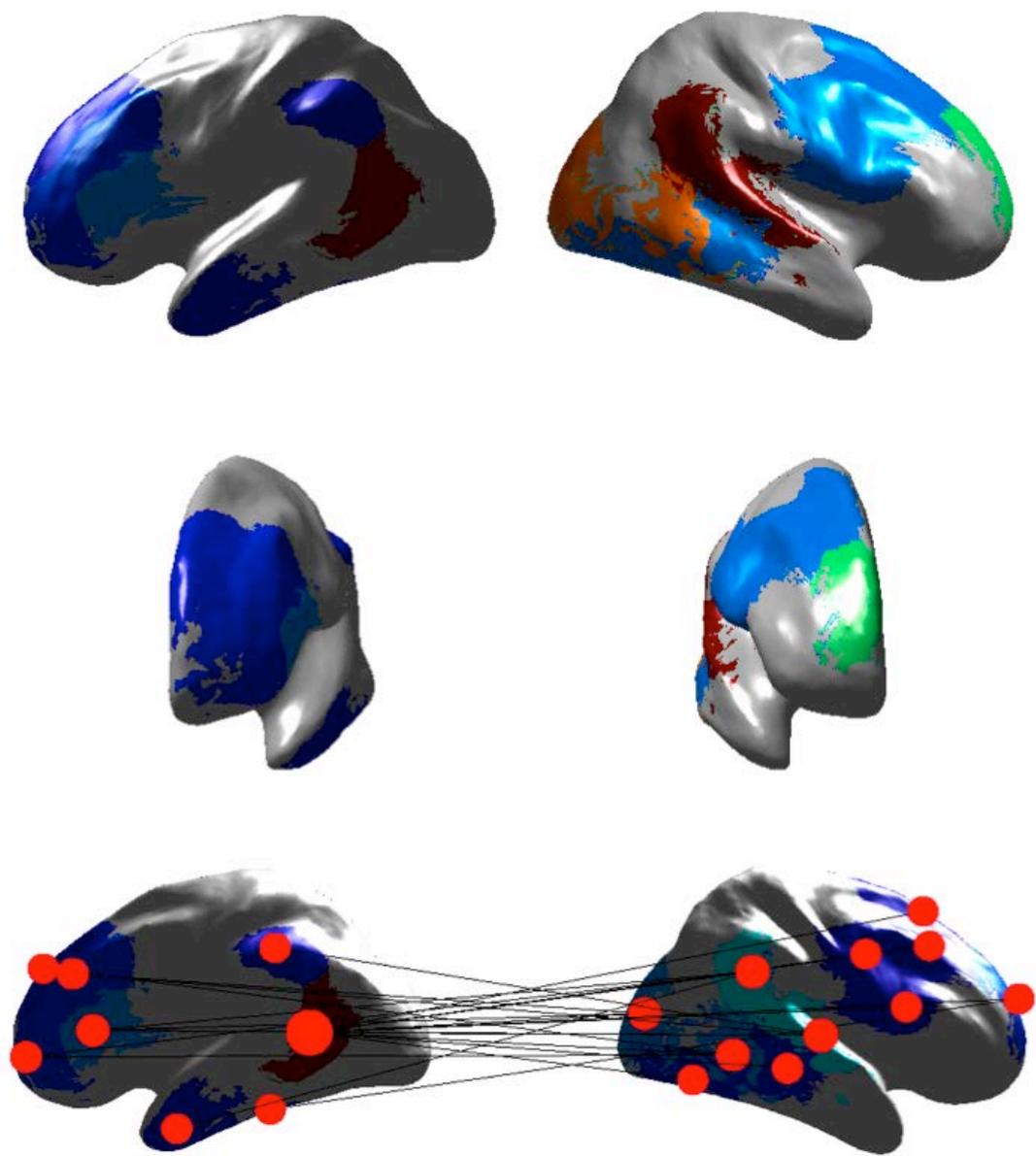


Figure 6-11 Theta Band Intra-hemisphere Networks.

In Figure 6-11, top and middle rows are displaying the degree centrality of significant cluster networks intra-hemispherically. Bottom row displays networks of statistically significant networks. Red circles correspond to centroids and their size is proportional to the number of connections. Black lines connecting circles represent significant cluster connections.



**Figure 6-12 Theta Band Inter-hemisphere Networks.**

In Figure 6-12, top and middle rows are displaying the degree centrality of significant cluster networks inter-hemispherically. Bottom row displays networks of statistically significant networks. Red circles correspond to centroids and their size is proportional to the number of connections. Black lines connecting circles represent significant cluster connections.

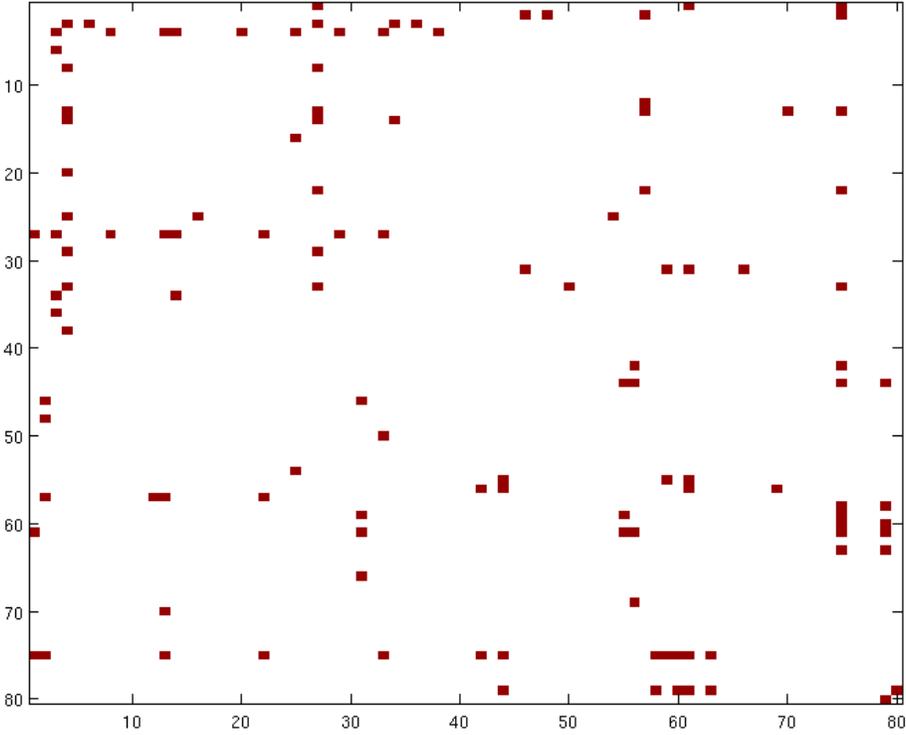
### **6.3.3 Alpha Band**

In Figure 6-13 the adjacency matrix of the alpha band network is shown. The alpha band consists of the frequencies 8-13 Hz. In Figure 6-14 the alpha band significant networks for intra-hemisphere are shown. The top four images show the degree centrality of the significant clusters of intra-hemisphere connections while the bottom image shows the network diagram for those connections. In the right hemisphere, the frontal cortex as well as premotor regions are heavily connected to visual areas in the occipital cortex. This is similar for the networks in the left hemisphere except for the additional contribution of temporal regions that are connected with visual areas as well as frontal areas.

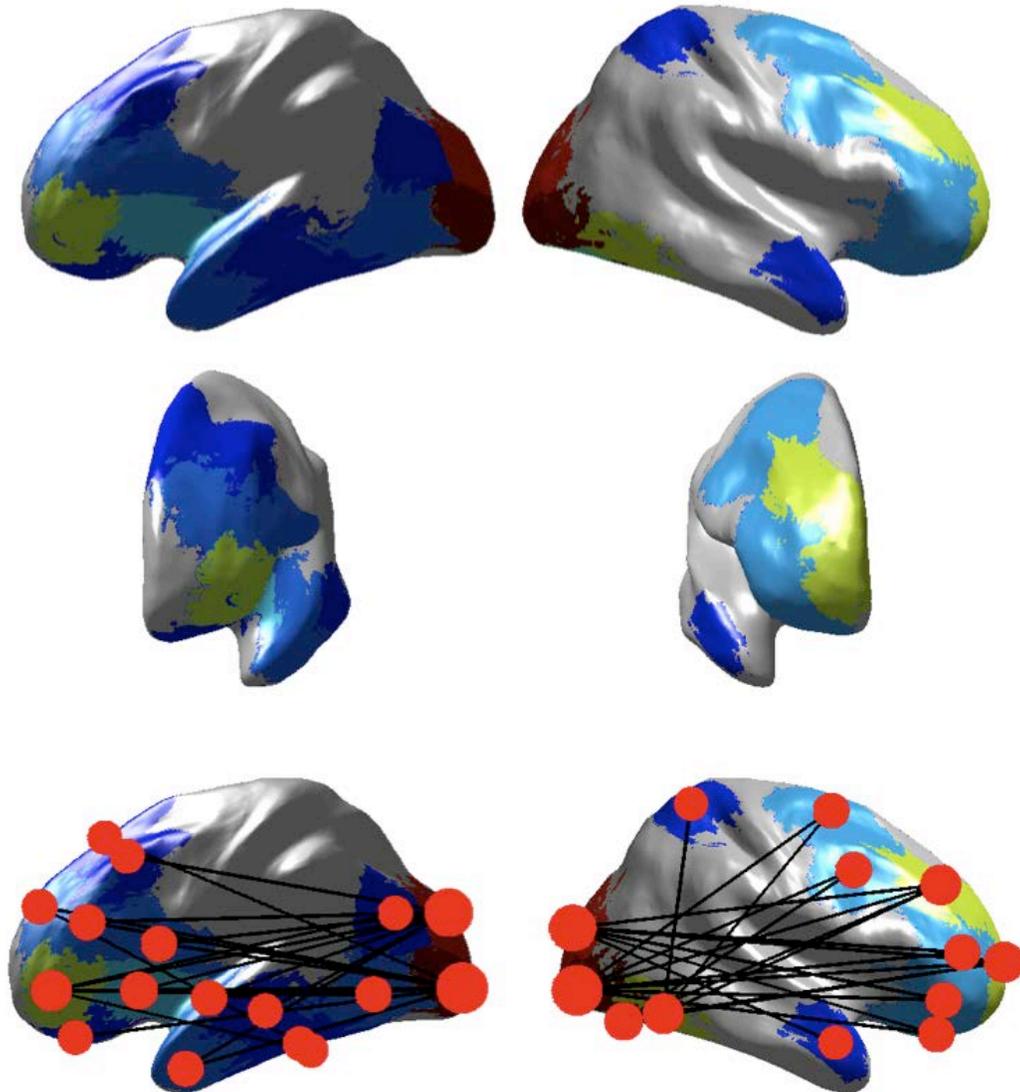
In Figure 6-15 the inter-hemisphere networks are shown for the alpha band network. Networks are heavily centered on the posterior portion of the lateral sulcus with additional contributions from portions of the dorsolateral prefrontal cortex. The visual cortex of the right hemisphere is also present connecting to a large number of areas in the left hemisphere including the left temporal lobe and the frontal cortex.

Figure 6-16 shows a subset of the inter-hemisphere connections in which the two seed regions are depicted (small brains in top left and top right corners) showing the seeded regions in red. These regions corresponded to the largest degree centrality in each hemisphere within the inter-hemisphere network connections. Regions corresponding roughly to the auditory cortex are

connected across the hemispheres. In addition, the region corresponding to the left posterior temporal lobe is connected to a number of regions in the right frontal cortex as well as portions of the ventral motor cortex. 40 clusters per hemisphere were used. The 80 cluster pairs are shown in the adjacency matrix. The first 40 correspond to clusters located in the left hemisphere while the last 40 correspond to clusters located in the right hemisphere.

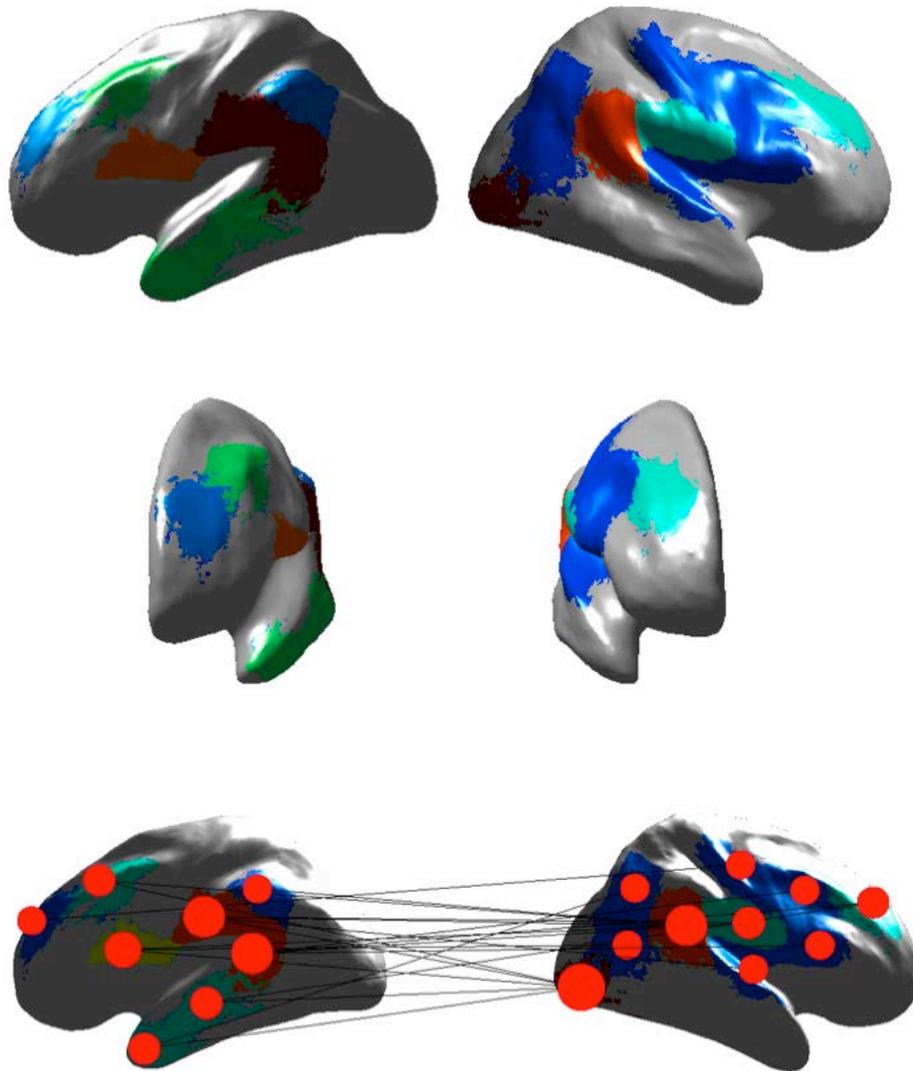


**Figure 6-13 Alpha Band Adjacency Matrix.**



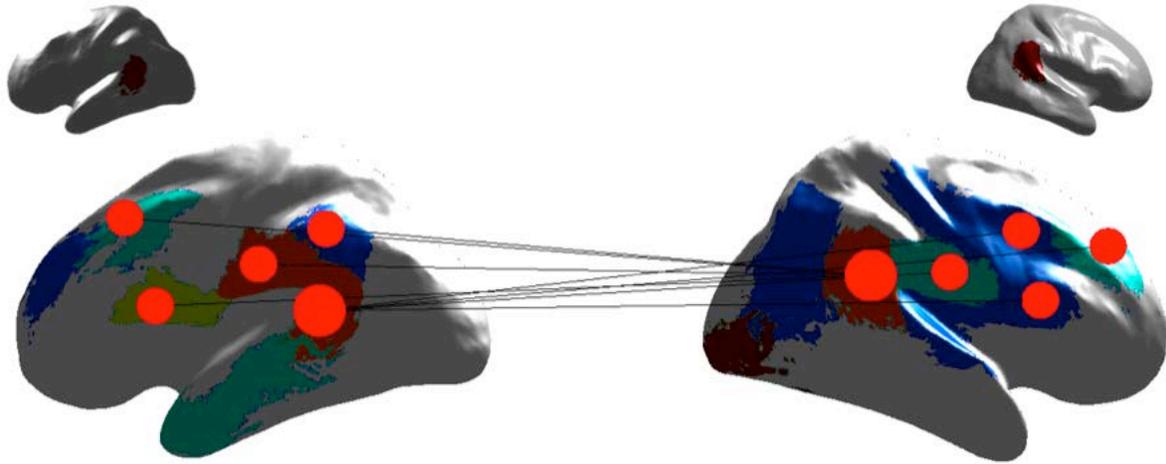
**Figure 6-14 Alpha Band Intra-hemisphere Networks.**

In Figure 6-14, top and middle rows are displaying the degree centrality of significant cluster networks intra-hemispherically. Bottom row displays networks of statistically significant networks. Red circles correspond to centroids and their size is proportional to the number of connections. Black lines connecting circles represent significant cluster connections.



**Figure 6-15 Alpha Band Inter-hemisphere Networks.**

In Figure 6-15, top and middle rows are displaying the degree centrality of significant cluster networks inter-hemispherically. Bottom row displays networks of statistically significant networks. Red circles correspond to centroids and their size is proportional to the number of connections. Black lines connecting circles represent significant cluster connections.



**Figure 6-16 Alpha Band Temporal Symmetry.**

In Figure 6-16, the largest degree centrality node of each hemisphere in the inter-hemisphere network were seeded on the surface. The seed regions are shown in the top left and top right corners as red areas. The network consisting of only those nodes which are incident with nodes on the opposite hemisphere are shown as a network in the larger dual-hemisphere brain. Red circles correspond to centroids and their size is proportional to the number of connections. Black lines connecting circles represent significant cluster connections.

### 6.3.4 Beta-Low Band

The beta-low frequency bands correspond to the range of 13-20 Hz. In Figure 6-17 the adjacency matrix for this cluster level network is shown. Figure 6-18, shows the degree centrality and the associated network diagram of intra-hemisphere significant network connections. The left hemisphere has a large number of areas significantly connected to at least one other area while the right hemisphere has significant connections in the frontal, temporal and occipital cortices.

Network connections in the right hemisphere generally connected frontal regions to occipital cortex locations. There is also a highly connected region in the parietal area that connects with locations in the temporal and occipital cortices and some areas of the ventral frontal cortex.

In the left hemisphere, connections between visual and frontal areas are dominant with large nodes occurring at the frontal cortex as well as the occipital cortex showing a dominant network effect in those locations. Connections between motor and premotor areas connect to regions within the visual and temporal cortices. Parietal areas are also connected with frontal regions of the cortex.

In Figure 6-19, the inter-hemisphere connections are shown with a dominant node located at the distal portion of the temporal lobe connecting to regions in the occipital, parietal as well as prefrontal areas on the contralateral side. Frontal areas in left hemisphere connect with regions in the occipital and parietal lobes of the right hemisphere. 40 clusters per hemisphere were used. The 80 cluster pairs are shown in the adjacency matrix. The first 40 correspond to clusters located in the left hemisphere while the last 40 correspond to clusters located in the right hemisphere.

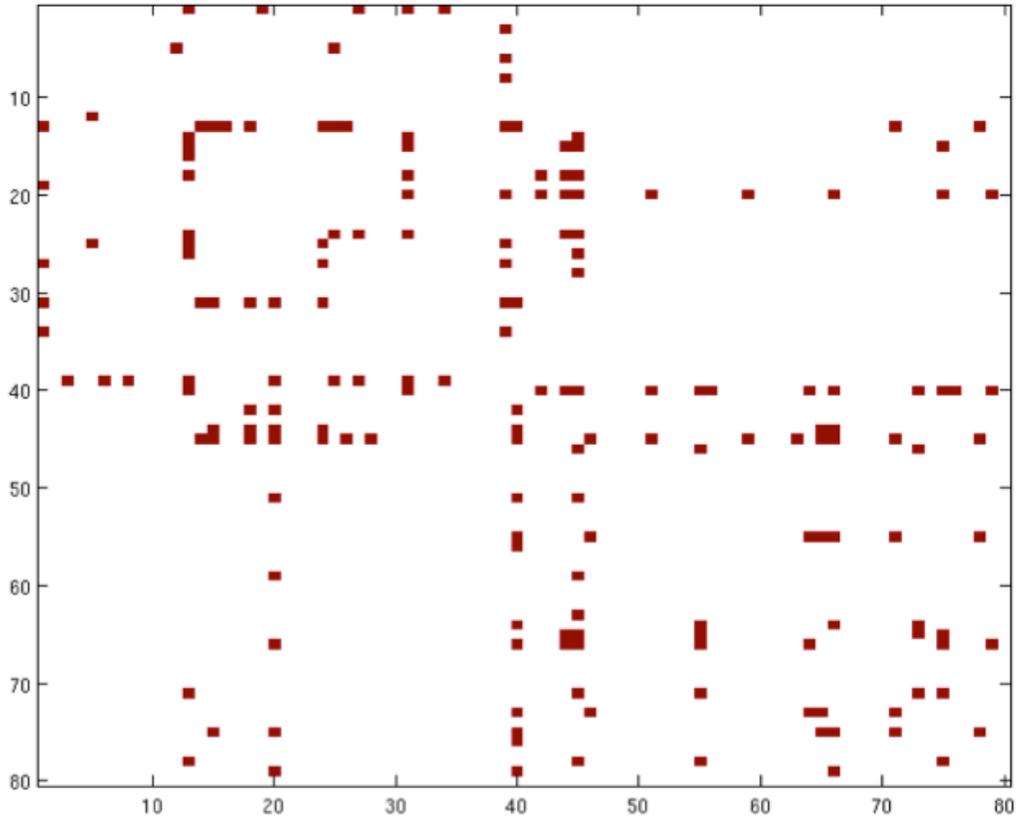


Figure 6-17 Beta-Low Band Adjacency Matrix.

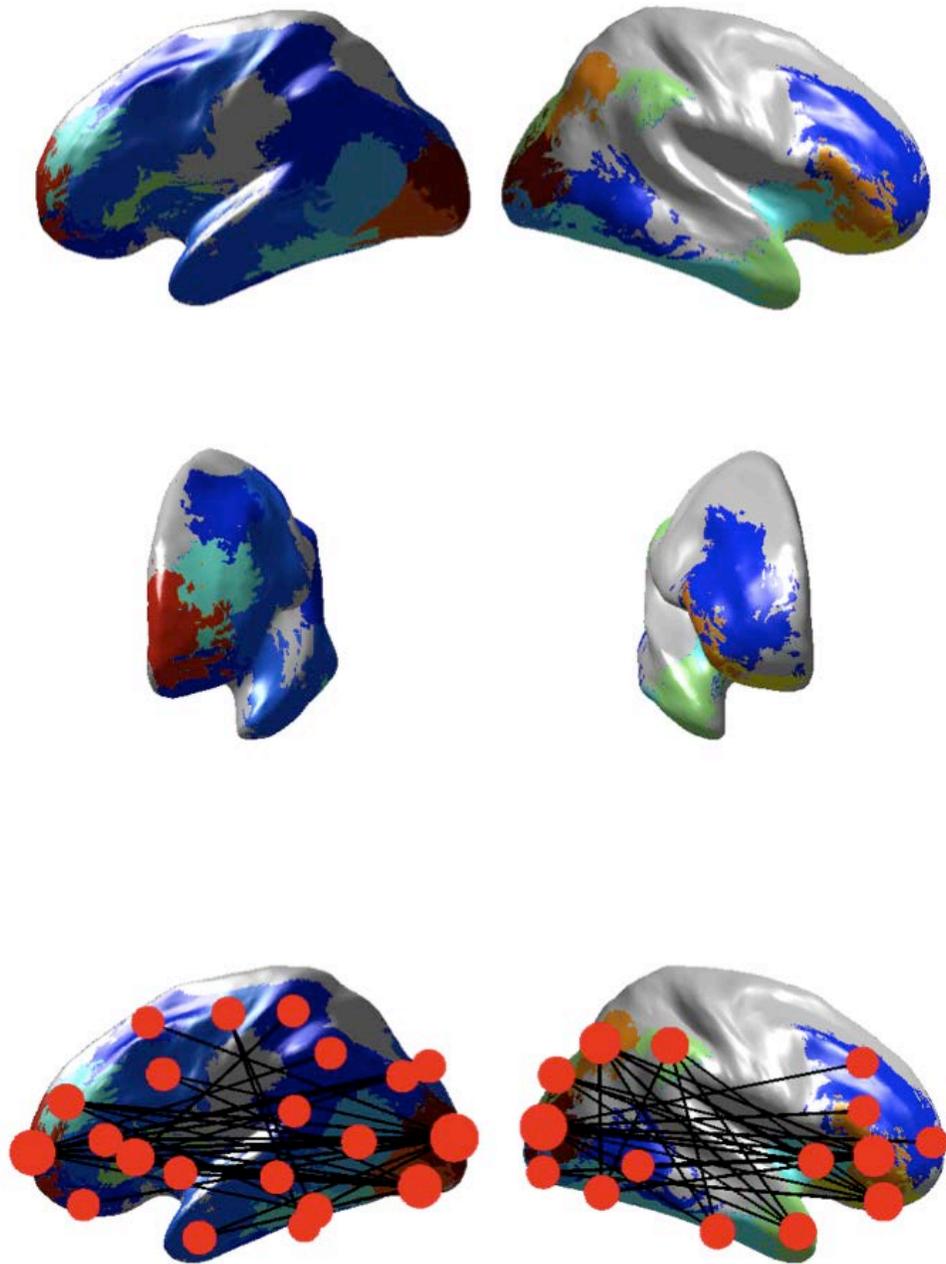
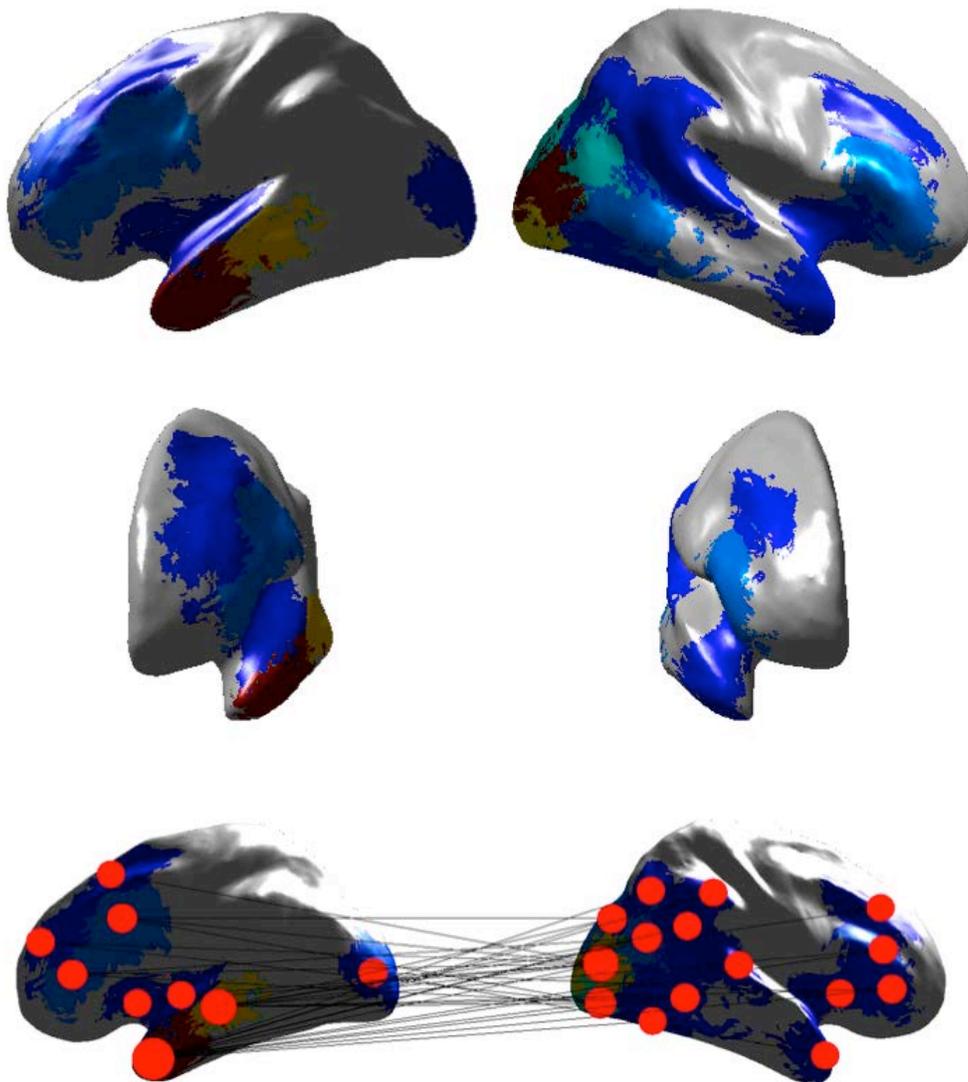


Figure 6-18 Beta-Low Band Intra-hemisphere Networks.

In Figure 6-18, top and middle rows are displaying the degree centrality of significant cluster networks intra-hemispherically. Bottom row displays networks of statistically significant networks. Red circles correspond to centroids and their size is proportional to the number of connections. Black lines connecting circles represent significant cluster connections.



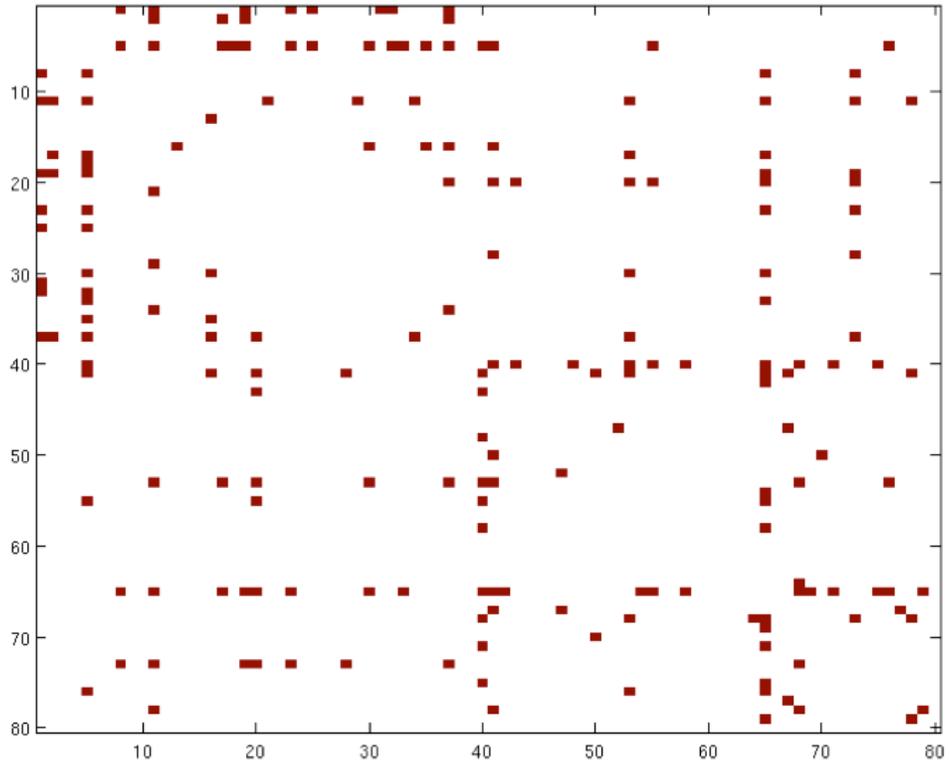
**Figure 6-19 Beta-Low Band Inter-hemisphere Networks.**

In Figure 6-19, top and middle rows are displaying the degree centrality of significant cluster networks inter-hemispherically. Bottom row displays networks of statistically significant networks. Red circles correspond to centroids and their size is proportional to the number of connections. Black lines connecting circles represent significant cluster connections.

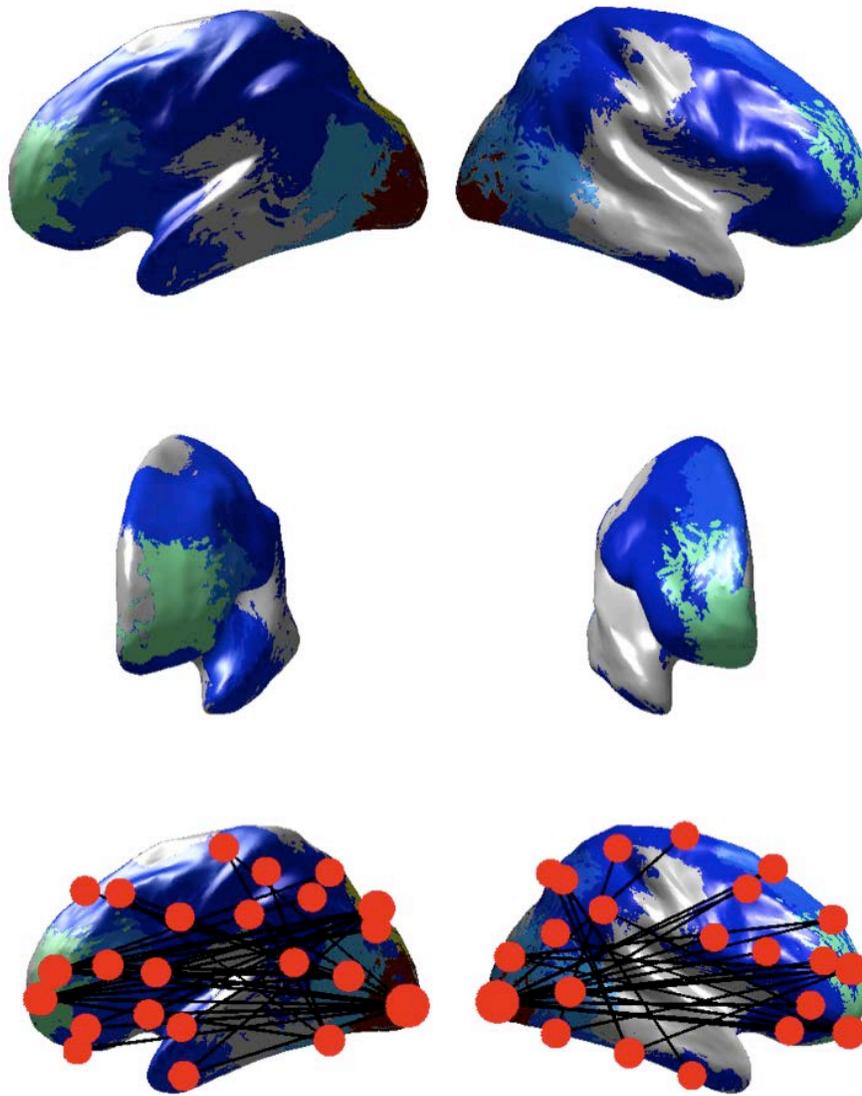
### **6.3.5 Beta-High Band**

The beta-high band corresponds to the frequency range of 20-30 Hz. The adjacency matrix for the beta-high network is shown in Figure 6-20. Significant intra-hemisphere network connections are shown in Figure 6-21. In both hemispheres, there are dominant connections from the visual cortex to frontal and premotor areas. Frontal-parietal as well as frontal-visual networks dominant the network connections.

Figure 6-22 shows the inter-hemisphere significant network connections in the beta-high frequency band. Node locations in the parietal areas of both hemispheres connect with many regions located in the occipital cortex and with frontal regions in contralateral sides. With the exception of a temporal lobe node on the left hemisphere, there is some symmetry between hemispheres. Forty clusters per hemisphere were used. The 80 cluster pairs are shown in the adjacency matrix. The first 40 correspond to clusters located in the left hemisphere while the last 40 correspond to clusters located in the right hemisphere.

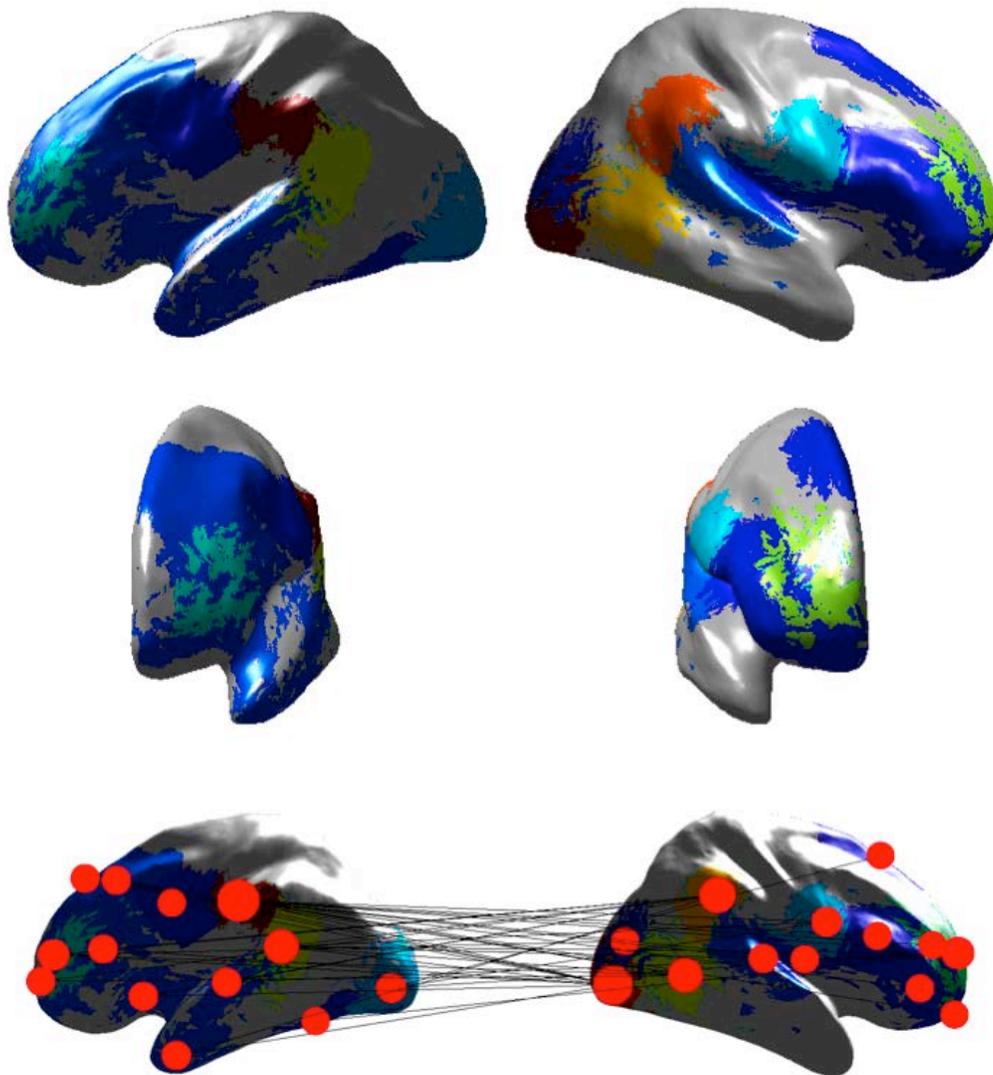


**Figure 6-20 Beta-High Band Adjacency Matrix.**



**Figure 6-21 Beta-High Band Intra-hemisphere Networks.**

In Figure 6-21, top and middle rows are displaying the degree centrality of significant cluster networks intra-hemispherically. Bottom row displays networks of statistically significant networks. Red circles correspond to centroids and their size is proportional to the number of connections. Black lines connecting circles represent significant cluster connections.



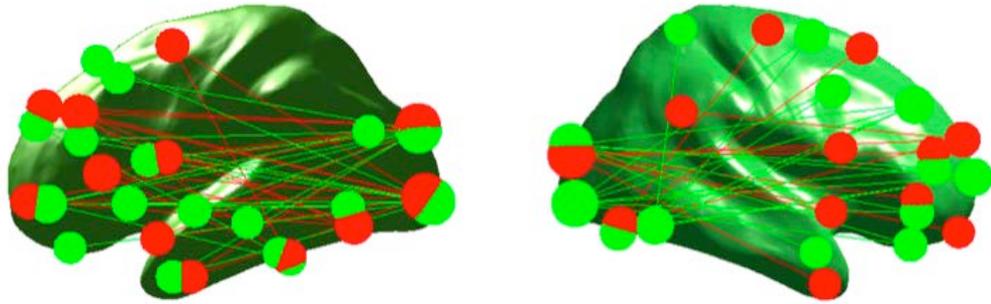
**Figure 6-22 Beta-High Band Inter-hemisphere Networks.**

In Figure 6-22, top and middle rows are displaying the degree centrality of significant cluster networks inter-hemispherically. Bottom row displays networks of statistically significant networks. Red circles correspond to centroids and their size is proportional to the number of connections. Black lines connecting circles represent significant cluster connections.

### 6.3.6 Inter-Band Results

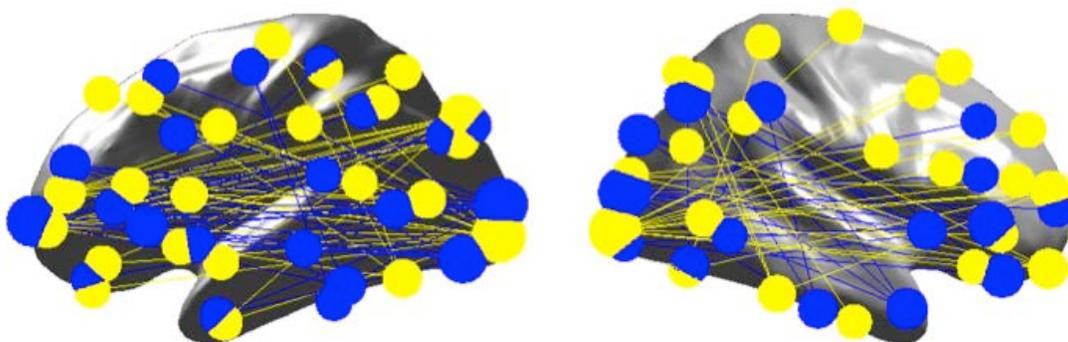
In Figure 6-23, the intra-hemisphere connections of the alpha (green) and theta (red) band frequencies are overlaid onto the same graph. Again, node size is proportional to degree centrality. Overlapping node locations are present where red and green nodes form incomplete circles as they overlap. Edge color corresponds to frequency band connection and matches node color.

Using the comparison, we can see that alpha and theta network have a high degree of overlap between node locations within the occipital and frontal cortices in both hemispheres. And in the left-hemisphere a high degree of overlap is seen in the temporal cortex. Frontal-visual networks dominant both connections with the additional recruitment of premotor areas present in theta band networks that connect to visual and temporal areas in both hemispheres. Statistically significant intra-hemisphere connections are overlaid between two frequency bands. Green circles represent significant nodes in the alpha band and green edges are significant connections between those nodes. Red circles represent theta band significant nodes and red edges are significant connections between those nodes. Node sizes are proportional to the degree centrality of that cluster within each frequency band.



**Figure 6-23 Alpha/Theta Intra-hemisphere Networks.**

In Figure 6-24, the intra-hemisphere connections between beta-high (yellow) and beta-low (blue) are shown similarly to Figure 6-23. A high degree of node overlap is seen in the frontal, temporal and visual areas and to some extent in the parietal area of the left hemisphere with some overlap occurring in posterior regions of the right parietal cortex. Differences in significant nodes are most noticeably seen in the right premotor and motor areas that exist in the beta-high frequency band but not at the beta-low frequency band. Frontal parietal and frontal visual networks dominant in both hemispheres at both frequency bands.



**Figure 6-24 Beta-Low/Beta-High Intra-Hemisphere Networks.**

Statistically significant intra-hemisphere connections are overlaid between two frequency bands. Blue circles represent significant nodes in the beta-low band and blue edges are significant connections between those nodes. Yellow circles represent beta-high band significant nodes and yellow edges are significant connections between those nodes. Node sizes are proportional to the degree centrality of that cluster within each frequency band.

## 6.4 DISCUSSION

Centrality measures are an indication of the relative importance of a node within a network. These values were used to cluster adjacent vertices into functionally defined seed regions on the basis of similar centrality scores in nearby vertices indicating a property of the network connection. Centrality scores are an important data reduction measure allowing for statistical analysis of the underlying connections between vertices. However, centrality measures also lend insight into the relative importance of that node with regard to the number of high importance nodes with which it connects. Therefore, our interpretation of centrality measures is as an indication of the importance of nodes.

The alpha band has high centrality scores at vertices corresponding to regions of the parietal and premotor areas. While we cannot interpret the specific location of vertices with which they connect, we can infer that those cortical locations have a large importance in facilitating the network of the underlying neural populations.

Subject specific centrality measures exhibit unique localizations on the cortical surface at each frequency band that is also present in the global averages of the frequency bands. The spatial smoothing present in the globally averaged centrality visualizations are the result of the transformations between subject specific anatomies and the global cortical surface. As a result of this interpolation procedure, activity in overlapping regions tends to become more focal and therefore we see smaller regions with higher centrality intensities in the global space. Despite this, patterns can still be seen in the specific anatomical locations of strong centrality measures at each frequency. These changing locations are likely the result of different cortical locations exhibiting different frequency-specific neural firing rates in synchrony with distant regions.

The averaged cortical centrality image across all frequencies exhibited a more spatially smooth left hemisphere indicating that the left hemisphere is on average more involved in connectivity than the right hemisphere with its relatively focal locations along the dorsal portions of the brain including the parietal and premotor areas. This would correspond to the non-dominant hemisphere of the subject handedness (recruited subjects were all right handed).

Each of the frequency band adjacency matrices provided a visualization of the relative densities between intra- and inter- hemisphere connections. The frequency band specific functionally defined cortical regions necessitated that each frequency band's cluster locations are uniquely ordered. Therefore, interpretation across bands must be done with visualizations overlaid upon the cortical surface.

To reduce the network complexity, each frequency band's networks were shown as inter- and intra- hemisphere networks. Across all frequency bands frontal-occipital networks were present in the intra-hemisphere networks. This is easily seen in the overlaid networks of the alpha and theta bands as well as the overlay of both beta bands. Changes in functionally defined regions are also seen in these overlays in which significant clusters are close on the cortical surface between the two frequency bands.

Due to the inter-subject variability a within band subject specific investigation would provide insight into the consistency of network dynamics across subjects. This might yield new insights into neurophysiological changes in network configurations. Additionally, some subjects exhibit inter-band similarities that are likely the result of the bandwidth of the wavelet kernel overlapping adjacent frequencies. For instance, at 20 Hz the two standard deviations frequency bandwidth of the Morlet wavelet is approximately 5 Hz. Therefore, the calculation of the 20 Hz frequencies actually corresponds to some instantaneous phase contributions from the beta-low and beta-high frequency bands. As a result, a given neural population may exhibit similar patterns in our band frequencies. This could be addressed by tailoring the frequency bands to those of the wavelet bandwidth exclusively to reduce overlap at wavelet windows greater than a set number of standard deviations from the fundamental frequency.

## **6.5 CONCLUSION**

Significant frequency band specific networks were visualized across the four physiological frequency bands. Network differences were shown at each of the specific frequency bands by both the differences in significant connections as well as differences in regions that connected to

other regions. Some similarities were present in dominant network connections however each frequency band exhibited different patterns of inter- and intra- hemisphere connections. Phase locking graph methods were shown to be a promising method for identifying and analyzing large-scale patterns of functional connectivity networks in resting state MEG.

## **7.0 WHOLE BRAIN FUNCTIONAL CONNECTIVITY USING DATA-DRIVEN PHASE LOCKING MEASURES OF RESTING STATE MEG**

### **7.1 INTRODUCTION**

Spontaneous functional connectivity (sFC) has become a critical tool for cognitive neuroscience. sFC is based on the analysis of the statistical relationships between spontaneous temporal fluctuations in brain signals between regions of the brain and has been proposed to reflect underlying neural communications between such regions. As a tool for neuroscientific research, sFC has provided insight to understand what the brain is doing outside of a task context [72, 94]. The use of sFC has become an important and widely used tool to examine functional connectivity in the typical brain [72], to chart neural development [95] and to characterize abnormal brain communication in a host of neurological and psychiatric disorders, such as stroke [96], Parkinson's disease [97], epilepsy [98], Alzheimer's disease [99], autism spectrum disorders [100], and is a major part of the Human Connectome Project [101].

Furthermore, studies suggest that abnormalities in spontaneous correlations are associated with neurological dysfunctions and may provide a biological marker for these disorders [96, 102-104]. In addition, recent studies have provided convincing evidence that the activity from neural regions that tend to co-localize with brain regions co-activated in response to a stimulus or task [105]. Thus, since sFC does not require the patient to perform any explicit task, this approach is of particular clinical interest in populations unable to comply with task instructions such as children, infants, sedated or comatose patients [2, 5, 35, 72, 73, 89, 95-98, 100, 104-110].

Although spontaneous fluctuations were first observed in blood oxygenation (BOLD) based functional MRI data by Biswal et al [5], similar networks have subsequently been observed in electrophysiological recordings of the brain by both electroencephalography (EEG) [6, 11] and magnetoencephalography (MEG) [2]. Many investigations have utilized functional magnetic resonance imaging (fMRI) due to the high spatial resolution [5, 9, 10, 111]. However, fMRI's low temporal resolution (0.5 Hz-2 Hz) and vascularly determined contrast does not lend itself to the direct investigation of high frequency neural cortical activity that have been proposed as a mechanism for information exchange between regions[112]. In addition, neural timescales are typically much faster than the recording capability of fMRI[38].

Magnetoencephalography (MEG) is a non-invasive measure of magnetic fields arising from dendritic activity in neural populations and is typically sampled at greater than 500 Hz allowing for direct investigation of fast neural activity [58]. Previous groups have used MEG for the investigation of complex cortical interactions utilizing resting state studies [2, 113, 114]. Studies have also demonstrated the efficacy of using phase relationships between cortical areas as a measurement of functional connectivity [36-38]. Specifically, the phase locking value (PLV) is a measure of the phase synchrony between two time-series, which has been previously applied

to resting state connectivity analysis in MEG [2]. As pointed out by Lachaux et al., phase-locking analysis is particularly well suited for connectivity analysis because it provides a measure of neural signal temporal relationships independent of their signal amplitude [36]. Here we employ PLV as a measure of frequency-specific relationships between cortical regions. This allows for studies that improve our understanding of both the spatial locations of network activity on the cortex as well as allowing us to directly investigate the relationship between frequencies and network communication.

Recently, analysis of sFC has begun to focus on the use of graph metrics to characterization of the spatiotemporal organization of connections and to quantify differences in these networks between patient groups or the brain states within an individual. Previous groups have demonstrated the efficacy of graph theory methods in quantifying these cognitive networks [75, 115, 116]. By employing network statistics, some studies have shown the ability of graph theory metrics to discern abnormalities in populations' network configurations and it has been suggested that dysfunctional cognitive networks may be indicative of wide range of neurocognitive deficits including Alzheimer's disease [45, 83, 116], drug addiction [117], schizophrenia [118, 119], as well as Huntington's disease [120]. These findings suggest the importance of identification and understanding of aberrant cognitive network functions that may lead to improvements in our understanding of disease processes. These electrophysiological recordings have provided a wealth of information about the cognitive processes that can be unlocked using data-driven approaches to identify important connections.

One of the challenges to analysis of data-driven methods on high-density electrophysiological recordings is the overwhelming dimensionality of the data. Computationally efficient and parallel computing methods have been developed and applied to a wide variety of

topics including the recent application to neuroscience datasets that leverage the use of data-mining and graph theory methods upon resting state connectivity data sets and has allowed for a richer data-driven analysis of spontaneous connectivity [4, 75, 116].

In this work, we have utilized parallel computing techniques to examine data-driven analysis of resting-state connectivity MEG data in an effort to provide computationally efficient solutions to the high-dimensionality of electrophysiological recordings. We present a method for quantifying simultaneous whole brain cortical network interactions by using graph theoretic metrics of network activity, specifically the eigenvector centrality. To capture multiple neurophysiologically relevant frequency bands, we used MEG because of its high temporal resolution. Following cortical reconstruction, we calculate the PLV between every pair of cortical dipole locations thus producing whole brain functional connectivity networks between every cortical surface location yielding a high-dimensional graph of network activity from which we apply non-parametric statistical methods to estimate the functional network activity. We show that these whole brain functional connectivity networks can be analyzed utilizing our method and further, we validate our method using numerical simulations. Finally, we apply our whole brain phase locking network methodology to real data and show that expected network patterns emerge with our data-driven approach.

## 7.2 MATERIALS AND METHODS

In this section, we will describe the implementation and testing of our method to characterize functional networks from all-to-all analysis of phase-locking connections from MEG data. We will first describe the overall architecture of our analysis and then provide evidence of its validity, characteristics, and limitations from numerical simulations. Finally, we will demonstrate this approach with application to experimentally acquired MEG resting state data.

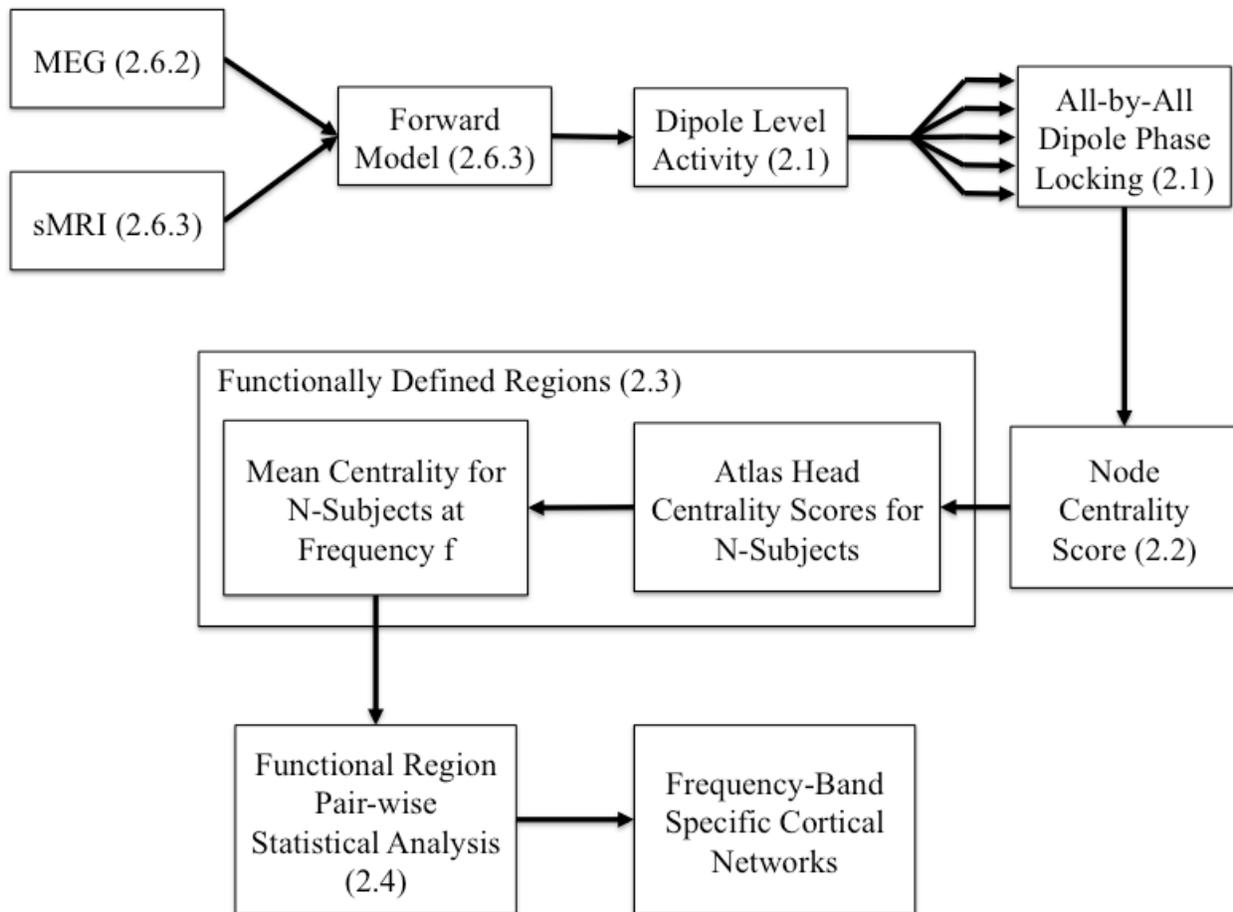


Figure 7-1 Overview of Method

Using inputs from MEG recordings and MRI structural information we reconstruct neural activity at each dipole on the cortical surface. The all-by-all dipole level phase locking is then calculated at the given frequency of interest for each subject before registration to a common atlas. Identification of functionally defined regions proceeds by clustering the centrality scores of the phase locking network in the atlas space. Using the regions as input, we then compute the all-by-all pairwise region-level network which yields the frequency band specific cortical networks.

A schematic outline of our proposed analysis stream is presented in Figure 7-1. In brief, after preprocessing of MEG data to remove external noise, eye-blink, cardiac, and motion artifacts (described in section 2.6.2), the MEG data from the sensors was reconstructed into brain (source) space using a minimum norm solution, yielding a movie of neural time-courses for each location in the reconstructed brain image as described in section 2.6.2. For each pair-wise combination of reconstructed dipole locations in the brain (in our example, 8,196 x 8,196 dipole locations), a time-averaged phase locking value was calculated for each temporal frequency of interest (details follow in section 2.1). The resulting all-by-all matrix describes the statistical coupling between all pairs of dipoles in the brain. Based on this matrix, the relative importance of each dipole is quantified using an eigenvector centrality score which describes the connectedness of each dipole to other brain regions (section 2.2). Eigenvector centrality is one of several centrality methods and was chosen because it assigns relatively high centrality scores to graph nodes that connect to other important nodes. Nearby dipoles on the surface of the cortex are expected to exhibit similar centrality scores because of both non-independence of the dipole estimates due to the nature of the MEG source reconstruction (inverse) problem and the intrinsic compartmentalized nature of the brain. Therefore, the next step of the analysis was to cluster nearby dipoles with similar scores into functionally defined regions (section 2.3). Finally, the

statistical probability of connections between regions is analyzed using non-parametric permutation testing to define the statistical graph of connected regions in the brain (section 2.4). The end result is a description of the functionally defined regions functional connectivity.

The key features of our proposed method include network characterization that leverages the pairwise phase relationships of the entire cortical surface as input, identification of regions determined via clustering of dipoles with similar network interactions amongst the population as well as utilizing a non-parametric statistic to account for the spatial correlations resulting from the MEG reconstruction method. These steps will now be described in more detail in the following sections. To characterize our model, we used both numerically simulated data and experimental data.

### **7.2.1 Resting State Phase Locking**

The application of phase locking analysis to MEG has been previously described in [2, 36, 38]. Phase locking is a measure of the propensity for two time series signals to maintain constant phase separation with each other over a period of time. Measures of resting state phase locking between time series electrophysiology recordings are an indication of the temporal variability of the phase differences between those time series at a given frequency. The explicit inclusion of frequency-specific information provides insights into networks as a function of frequency.

To calculate the phase locking value, two time series are first spectrally decomposed at a given frequency,  $f_0$ , to obtain an instantaneous phase estimate at each time point. Calculation of instantaneous phase was performed using a continuous wavelet transform [38]. A Morlet wavelet was chosen because it is Gaussian shaped in both the temporal and spectral domains. The Morlet wavelet is described by the following equation:

$$w(t, f_0) = (\sigma_t \sqrt{\pi})^{-1/2} e^{-t^2/2\sigma_t^2} e^{2i\pi f_0 t} \quad (7-1)$$

The spectral bandwidth,  $\sigma_f$ , is then obtain via  $\sigma_f = 1/(2\pi\sigma_t)$  and a constant ratio enforced via  $f/\sigma_f = c$  where we choose  $c = 7$  consistent with Tallon-Baudry et. al [38], because it provides good spectral resolution at lower neurophysiological frequency ranges while still maintaining resolution at higher frequencies. Our target for neurophysiologically relevant frequencies of interest was integer values between 5-40 Hz and we choose our value for  $c$  so that it provides good resolution at these frequencies. Characteristic of wavelets, the temporal and spectral bandwidth is a function of the frequency. As an example, at  $f = 10$  Hz the  $\sigma_t = 111$  ms and  $\sigma_f = 1.4$  Hz and similarly at  $f = 30$  Hz then  $\sigma_t = 37$  ms and  $\sigma_f = 4.3$  Hz showing that the temporal and spectral bandwidth of the wavelet are a function of the center frequency.

After obtaining instantaneous phase estimates the time-averaged phase locking value can be computed via the following equation:

$$PLV = \frac{1}{N} \left| \sum_{n=1}^N e^{i(\theta_1(n) - \theta_2(n))} \right| \quad (7-2)$$

where  $N$  is the number of sampled time points and  $\theta_1$  and  $\theta_2$  are the instantaneous phase values at time point  $n$ . Phase locking values range from 0 for a random phase relationship to 1 for a fixed phase relationship. Phase locking is an un-directed measure and is therefore symmetric (i.e.  $PLV(s_1, s_2) = PLV(s_2, s_1)$ ).

Whole brain phase locking networks are computed between each pair of time series measurements on the cortical surface. Due to the symmetry of phase locking, we need only compute one phase locking value per pair of time series. For  $L$  total cortical locations, we must compute  $L(L-1)/2$  phase locking values to obtain the all by all phase locking network.

### 7.2.2 Dipole Centrality

The whole brain phase locking network are represented using graph theory in which vertices correspond to cortical dipole locations and phase locking values between vertices correspond to weighted edges between those vertices [4, 121]. The centrality of a graph is a measure of the relative importance of a node and is used to characterize networks. There are several variations of centrality metrics and in this work, we chose to apply eigenvector centrality (EVC) because it incorporates the entire graph structure in determining the relative importance of each node in the network. EVC is used to characterize a node in a network according to the amount of connections that node has with other important nodes [122, 123]. Therefore, in phase locking networks, it provides an estimate of the importance of a single dipole within the whole brain network.

The eigenvector centrality of the network was obtained by computing the first eigenvector after applying the singular value decomposition of the adjacency matrix of the phase locking network [124], where the adjacency matrix is the  $L \times L$  square matrix in which each entry,  $a_{i,j}$  corresponds to the PLV value between the  $i^{\text{th}}$  and  $j^{\text{th}}$  dipole location. This yields a vector of length  $n$  in which each entry,  $d_i$ , is the relative importance of the  $i^{\text{th}}$  dipole. For two dipoles in the  $n$ -length centrality vector,  $d_i$  and  $d_j$  in which the values of the two entries are such that,  $d_i > d_j$ , then we say that the  $i^{\text{th}}$  dipole is more important than the  $j^{\text{th}}$  dipole within the network. The diagonal elements of the phase locking adjacency matrix, which correspond to phase locking value between a time series and itself, are set to zero corresponding to non-self-referential edges (i.e. dipoles do not form network connections with themselves).

### 7.2.3 Anatomically biased functionally defined regions

Functionally defined regions were calculated by clustering for spatially contiguous cortical surface dipole locations that contained similar patterns of centrality scores. This is in contrast to using an independent anatomical localizer [125-127]. The identification of functionally defined regions was motivated by the need to identify starting regions from which to initialize statistical methods for identifying region-based clustering. To that end, we used a clustering algorithm that incorporated both the spatial location as Cartesian coordinates of the dipole on the cortex as well as the eigenvector centrality scores of each of those dipole locations.

This clustering method was chosen such that regions were spatially smooth along the cortical surface to prevent disjoint functional areas. We used the distance along the manifold surface of the cortex to constrain clusters of neighboring points. In this way, dipoles along the same gyrus would be more related than locations spanning across a sulcus, thus better matching our expectation about the compartmentalization of the brain. In order to do this, we used coordinates in a spherically registered representation of the cortex as partial inputs to the clustering algorithm. As described in section 2.6.3, MEG source localization was performed based on anatomical information from MRI data, which was processed and segmented using the FreeSurfer software [128, 129]. The final stages of FreeSurfer analysis involve the extraction, inflation, and registration of the tessellated cortical surface into a spherical space, which forms the basis for dipole locations. We used a cosine distance metric for calculating the distances between vertices so that regions were contiguous by minimizing the angle between them. Cosine distance is given by the following equation:

$$distance(A, B) = \cos(\gamma_{A,B}) = \frac{A \cdot B}{\|A\| \|B\|} \quad (7-3)$$

where  $\gamma_{A,B}$  is the angle between the two vectors  $A$  and  $B$  that are n-dimensional vectors from the k-means algorithm and correspond to  $A = [x_a, y_a, z_a, EVC_a]$  with first three components being the surface coordinates on the brain surface, and where  $\|A\|$  indicates the Euclidean norm of the vector  $A$ .

Using k-means clustering allowed us to compute cluster membership of the four-dimensional input space [130]. K-means yields  $k$  clusters from the original data set and attempts to minimize the equation:

$$arg \min_S \sum_{i=1}^k \sum_{x_j \in S_i} distance(x_j, \mu_i) \quad (7-4)$$

where  $S$  is the configuration of the clusters,  $x_j$ , is the location of each original dipole location and corresponding EVC value,  $\mu_i$  is the centroid mean at the  $i^{\text{th}}$  cluster, and the distance function assigns a scalar value to the distance between  $x_j$  and  $\mu_i$ . The heuristic attempts to find the best configuration of  $S$  such that we minimize the errors.

To calculate the k-mean clusters we employed an iterative solver from MATLAB (MathWorks Natick, MA) that uses a heuristic algorithm [130]. The solver works by first assigning k random data inputs as centroid locations. The iteration then proceeds in two steps. First, each point is assigned to its closest centroid according to the distance metric. Then, each cluster's centroid is recalculated given the new input values corresponding to that cluster and the closest data point to the new centroid is assigned as the centroid of that cluster. This method continues until convergence is reached in which the centroid assignments are unchanged between iterations or the maximum number of iterations is exceeded. We choose a maximum number of iterations for solution convergence at 1,000 iterations.

When choosing the number of clusters, k, for use in the k-means algorithm there is an inherent tradeoff between computational complexity and the homogeneity of clusters. For increasing values of k, the number of permutation tests that must be run increases as the square of the number of clusters while the spatial extent of each cluster decreases. This leads to increases in the complexity of the calculation making it more difficult to achieve significance, however smaller clusters would yield greater similarity between dipoles contained in the cluster and provide more granular detail of functional regions. To ascertain an appropriate k-value, we calculated a dendrogram in which we iteratively generated clusters using successively smaller k values upon the obtained centrality scores [**Error! Hyperlink reference not valid.**]. Briefly, at each iteration the number of clusters is reduced by one and correspondingly two clusters are aggregated into a combined cluster with a lower similarity score (as a result of inhomogeneity of the precursor clusters). We can characterize this level of dissimilarity by computing 1-similarity between the merged clusters. The choice of  $k = 40$  corresponded to cutoff in which we chose the number of clusters small enough to be computationally tractable while still providing

neuroscientifically relevant network structures such that clusters were small enough to make interpretation possible. This value was used for both the real and simulated data. The dendrogram is shown in Figure 7-2, where increasing y-axis results in greater dissimilarity between clusters being joined at lower levels. The red dashed line in Figure 7-2 indicates our choice for the number of clusters.

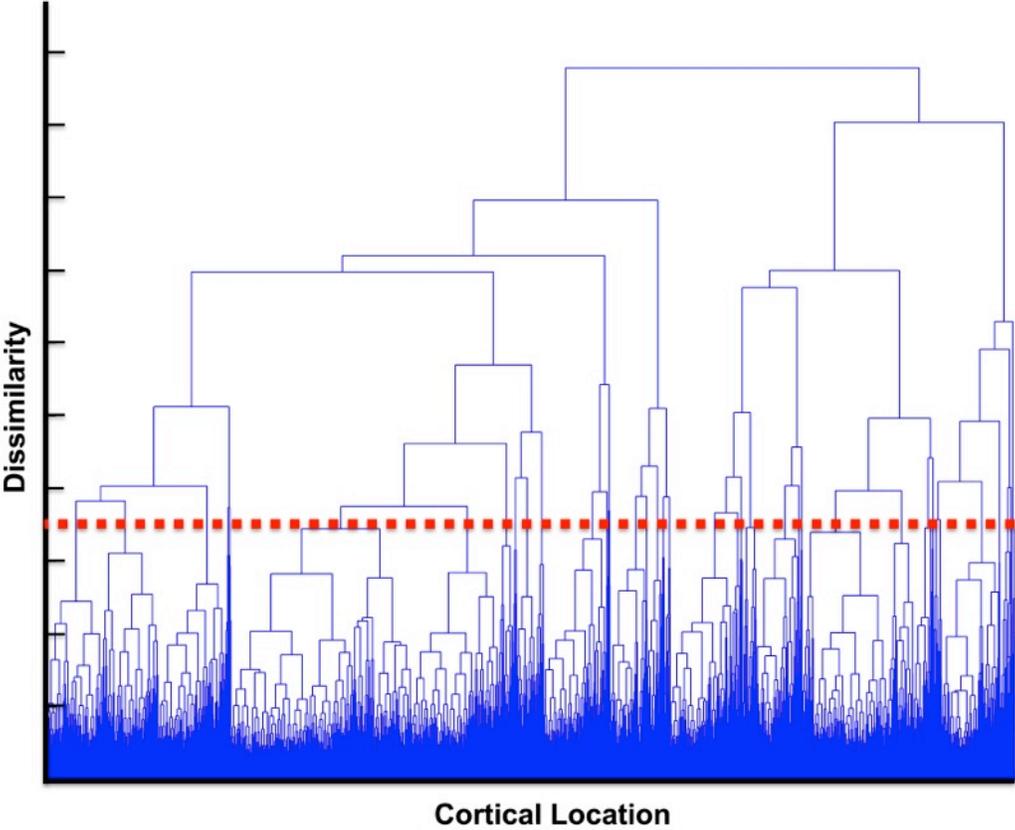


Figure 7-2 Dendrogram

#### 7.2.4 Statistical testing of regional connections

The final step of our approach was to perform statistical tests on the connections between functionally defined regions. We determined significant connections between these functionally defined regions via a non-parametric permutation test. Each pairwise grouping of functional regions was compared across all subjects in the resting state scans to the empty room scans. To calculate statistically significant connections between pairs of regions we used a non-parametric permutation that explicitly tests the underlying distribution of dipole level phase locking values between the two functionally defined regions.

The permutation test follows the method described in Maris et al. [51]. For  $S$  subjects, the permutation test is as follows for a given pair of regions. First, the phase locking values are obtained for those PLV values that connect the two regions (i.e. edges that connect the two regions). This population is aggregated over the subjects to obtain a distribution of phase locking values by binning PLV values to create a histogram. We obtained this distribution for both the resting state and empty room scans. The null hypothesis in our permutation test is that the empty room scan data and the subject recording are the same. In permutation tests, any metric may be chosen and here we used the difference in means between the two populations as our test statistic because of its low computational overhead [51]. Our observed test statistic is therefore the difference in the mean between the two populations of empty room and resting state data.

Next, we performed permutation of the data in the two populations, empty room and subject data, and applying our test statistic to every permutation of the combined data. From this permutation calculation we obtained a distribution of test statistics. At a given critical value,  $\alpha$ , we found the test statistic that corresponded to that critical value. Finally, we tested if our observed test statistic is more extreme than our critical value. If the observed test is greater than

the critical value, we rejected the null hypothesis that the two groups come from the same distribution. This indicates a statistically significant connection between the two functional regions and therefore a positive connection between the two clusters.

For  $k$  functionally defined regions obtained via our clustering method, there are  $k(k-1)/2$  pairs of regions on which to perform this permutation test. In addition, due to the large number of dipole connections between two regions, we used a Monte Carlo (MC) sampling of the permutation test to reduce the number of permutations required by iteratively sampling the permutation space until successive iterations yielded a small change in outcome the distribution of test statistics. For our stopping criteria, we required a minimum of 5,000 permutations after which we stopped the MC algorithm when there was less than a 0.01% change in the distribution. This method has been shown to asymptotically equivalent to the full permutation test [53, 54]. Following computation of all pairwise connections at the functional region level, we obtain a graph network of connections between functional regions at the given frequency.

### **7.2.5 Description of Numerical Simulations**

In order to explore the performance and sensitivity of our approach, we first generated a set of synthetic simulation MEG data. The purpose of these simulations was to quantify the sensitivity and specificity of the proposed methodology. In particular, the estimation of source space signals in MEG introduces a spatial blurring and subsequent point spread function due to the MEG reconstruction procedure. This effect has been shown to introduce false-positive correlations between nearby cortical surface points [2]. In order to evaluate the rate of false positives generated by the spatial point spread, we conducted simulations at nine dipole pair locations to examine our method's ability to recover the original signal locations. We performed

numerical simulations in which dipole signals were generated and projected into MEG sensor space. We applied our proposed method to test the differences between the simulated dipole information and the networks produces via our method.

Simulations of source space signals were conducted utilizing the MNE software package using a method similar to Ghuman et al. [2]. Briefly, simulations were conducted by generating 10 Hz sinusoids at two dipole locations with the same frequency at each location that, following the addition of noise, are able to provide a realistic simulation of subject data. One dipole location was fixed on the cortical surface within the parietal lobe and for each of the nine simulation distances the location of the second dipole was moved incrementally closer starting in the frontal lobe so that we could evaluate the effect of inter-dipole distance on our method. Inter-dipole distances ranged from a 10 cm to 1 cm. In sensor space, noise was added according to a covariance matrix obtained from empty room recordings. Five minutes of data were generated for each simulation at 250 Hz with 306 channels in MEG sensor space.

The sinusoidal amplitude was determined by matching the PLV values obtained in real data. First, we calculated the average PLV in subject data at the distances used in our simulation. Next we set the amplitude of the sinusoidal generators such that the PLV at the equivalent distances had the same PLV following our simulation procedure. Following the addition of noise and the reconstruction onto the cortical surface, the original sinusoids were contaminated with realistic noise in sensor space data. The sinusoidal amplitudes we used ranged from  $1.5 \times 10^{-8}$  at the largest distance to  $3.5 \times 10^{-7}$  at the shortest distance. This has the effect of fixing the SNR of the simulation to be comparable to PLV present in real data by matching the distribution of PLV signals (this corresponded to a simulation SNR of -26 dB). A total of nine positions were used in simulations. For each dipole position pair, 10 repetitions were performed.

Due to regularization of the MEG source localization problem, the resulting images will be spatially blurred. In order to account for the point-spread function of the inverse problem in the estimate of the false-positive rate of the model, sensitivity and specificity analysis was performed by two alternative definitions of positive clusters in which two clusters were statistically connected. First, as those clusters that contained the original dipole and second as a more relaxed criteria of clusters which were located within a specific distance of the dipole of interest (i.e. the dipole used as a sinusoidal generator) based on the full-width half maximum (FWHM) extent of the reconstructed dipole in empty room data.

For each dipole location we calculated the full width half maximum (FWHM) radius around the given dipole in Cartesian space. We began by calculating the distribution of inter-dipole phase locking values from the dipole of interest to every other dipole and ordering the distribution by distance between each dipole pair. One distribution was obtained for each of the nine dipole positions used in the simulation. We then calculated the FWHM for each position of the distributions by determining the inter-dipole distance that corresponded to half of the maximal inter-dipole PLV value. For each dipole used in the simulation (black dot) the FWHM is calculated (red circle) as list of points on the cortical surface within that radius. Clusters that are incident within the FWHM radius are part of the FWHM true positive rate.

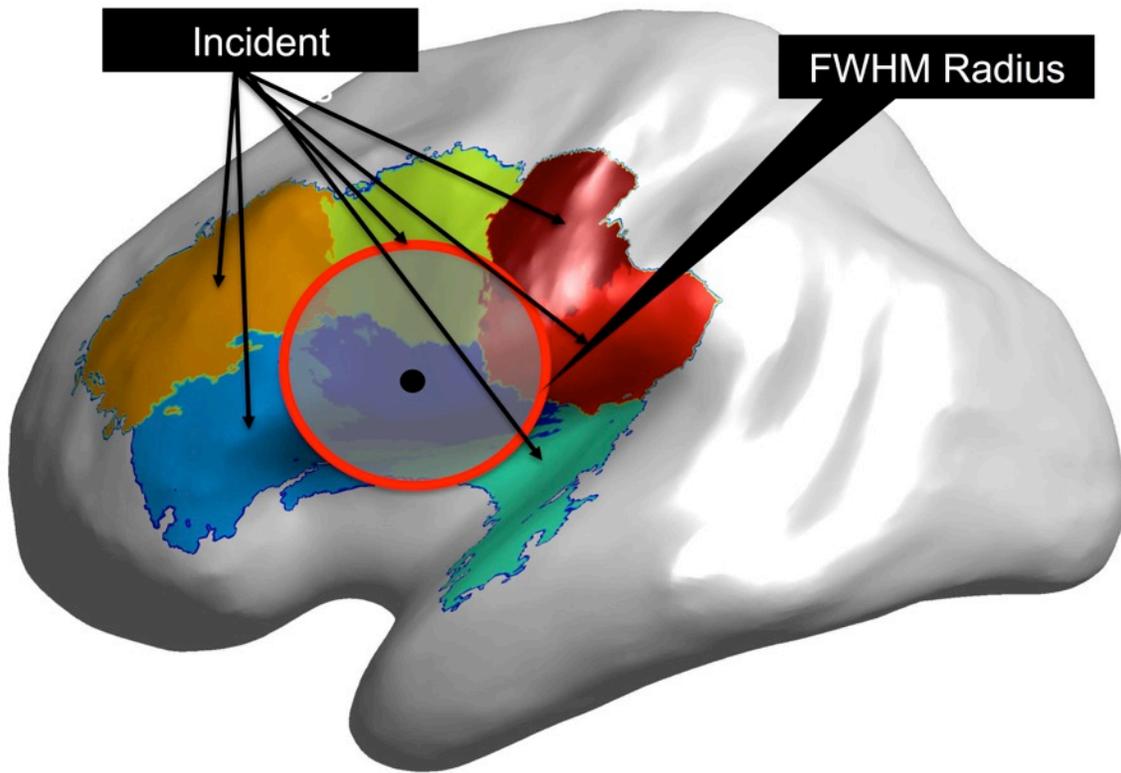


Figure 7-3 Calculation of Full Width Half Max (FWHM)

For each cluster, we computed whether any of the dipoles that were members of that cluster fell within the FWHM distance from the simulation point. Clusters that possessed this property were considered to be within the FWHM radius of the around the dipole location. A representation of this process is shown in Figure 7-3 where the source dipole location is shown and an accompanying overlay of the clusters with surfaces that contain areas that are within the FWHM radius. This provided an additional metric for understanding the role of reduced resolution resulting from the point spread function. Namely, we are able to test the resolution of the original signal generating dipole as a function of distance from that dipole.

### **7.2.6 Evaluation of PLV Characteristics**

In order to characterize the robustness of our method, we characterized the effects of signal modulations on our ability to resolve significant connections between network clusters. We simulated a 306x306 two-dimensional grid of locations each with an individual time series. Each time series in the grid included iid Gaussian noise with the same SNR from real subject data. At two fixed 10x10 grid locations we generated a sinusoidal signal, similar to our above simulation, to simulate brain activity in those areas. In this reduced model, we applied our methodology to identify clusters of activity and generate significant phase locking networks. To understand the effect of different signal modulations on our ability to resolve connections, we included four perturbation conditions of the two simulated brain activity regions. Using this model, we tested how different signal manipulations affect the network graphs.

The four manipulations consisted of SNR variations, signal modulation, phase variances and frequency offsets. First we modified the signal strength of the sinusoid by modifying the amplitude of the sinusoid to adjust the SNR relative to the real data ( $2.0 \times \text{SNR}_{\text{real}}$ ,  $1.0x$ ,  $0.5x$ ,

0.1x amplitude multiples of amplitude used in the above simulation). To test signal modulations, we included a low-frequency carrier signal and tested at multiple low-frequency modulators (0.1 Hz, 0.5 Hz, 1, Hz, 5 Hz). We also created phase variances and frequency offset simulations. The signals in the two regions were generated according to the following equations:

$$\begin{aligned} s_1(t) &= \sin(w_1 t + \phi_1) + \varepsilon \\ s_2(t) &= \sin(w_2 t + \phi_2) + \varepsilon \end{aligned} \tag{7-5}$$

where  $s_1$  and  $s_2$ , are used to generate signals in the active regions,  $w$ , is the frequency of the underlying sinusoid and  $\varepsilon$  is iid Gaussian noise. Phase variances were investigated by setting  $w_1 = w_2$ , and allowing  $\phi$  to be a uniform random distribution with mean zero. That is, phases randomly drawn from a uniform distribution on the range  $-\pi$  to  $\pi$  (values of  $M$ :  $\pi/10$ ,  $\pi/4$ ,  $\pi/2$ , and  $\pi$ ). Finally, to understand our methods' ability to differentiate different frequencies, we set  $w_1 \neq w_2$  while fixing the phase variance with  $\phi = 0$ . We set  $w_1 = 25$  Hz and varied  $w_2$  at 20 Hz, 15 Hz, 10 Hz, 1 Hz. Each of the perturbation settings were repeated 50 times producing a total of 800 simulations (4 conditions x 4 settings x 50 simulations). The results of the simulations were analyzed using a receiver operating characteristic (ROC) of the true positive versus false-positive significant cluster connections. We defined a true positive to be a significant connection between the two clusters containing the active regions regardless of the perturbation condition. This definition allowed us to compare the methodologies resolution across conditions and perturbations.

## **7.3 EXPERIMENTAL METHODS**

### **7.3.1 Subject population**

Resting state (spontaneous) MEG signals were recorded from five healthy human subjects (3 males, ages 25-45 and 2 female, ages 20-25). All subjects were right-handed. Each subject participated in two 5 minute resting state MEG scans which were concatenated together in source space prior to pre-processing. In each scan the subject was instructed to fixate on a centrally placed cross-hair, eyes open while sitting upright. Each scan lasted five minutes. Prior to subject scanning an empty room dataset was collected which consisted of 10 minutes of MEG data collection without a subject present in the MEG. The institutional review board of the University of Pittsburgh approved all study procedures. Written consent was obtained from each participant prior to participation in this study.

### **7.3.2 MEG recording and preprocessing**

A 306-channel Elekta Neuromag scanner was used for recording of neuromagnetic signals. Head position coils were placed on the scalp to determine head position during recording. Temporal spatial signal separation (tSSS) was employed to remove external magnetic field contributions in the signal recordings which can lead to spurious signals which appear to have originated from physiological changes but are in fact artifacts external to the MEG sensors [93]. Time series

MEG recordings were preprocessed to remove excess noise and artifacts prior to connectivity analysis. The MEG signals were first bandpass filtered using cutoffs of 1 Hz and 50 Hz to remove the 60 Hz line noise and slow drifts. MEG scans were sampled at 1000 Hz and subsequently downsampled to 250 Hz.

### 7.4.3 Structural MRI and MEG Source localization

Structural MRI images were obtained for each subject using a 3-Tesla whole body MRI scanner (Siemens). A T1-weighted brain volume was recorded for each subject. Following coregistration between MEG fiducial markers and the MRI structural images a cortical surface model was obtained using the FreeSurfer™ software package. A topologically accurate cortical surface mesh consisting of approximately 150,000 vertices per hemisphere were subsequently downsampled to 4,096 vertices per hemisphere for use in inverse reconstruction.

Reconstruction of cortical dipole activity from MEG sensor measurements is an ill posed problem but an estimation procedure has been described . Briefly, we can estimate this reconstruction by utilizing the minimum norm estimator using the MNE software package. Our data model is given by:

$$\hat{y}(t) = Wx(t) \tag{7-6}$$

where  $x(t)$  is a time series of MEG sensor space recordings and  $\hat{y}(t)$  is a time series of dipole location estimations. The minimum  $L_2$  norm solution for  $W$  is then given by:

$$W = RA^T(ARA^T + \lambda^2C)^{-1} \quad (7-7)$$

where  $A$  is the gain matrix of the forward problem and  $\lambda$  is a regularization parameter and is given by  $\lambda^2 = 1/SNR$ .  $C$  represents the covariance of the noise in the sensors and  $R$  is the covariance matrix of the sources [2, 133]. The depth dependent decay of MEG signals was accounted for by including a depth factor of 0.8 onto the  $R$  matrix [134]. Furthermore, due to the typical orientation of cortical neurons to be normal to the surface an orientation constraint was applied to  $R$  such that those components that are tangential to the surface were multiplied by 0.4 and those normal to surface were multiplied by 1 and finally only the normal component is taken. Following the procedure from Ghuman et. al, we calculated the noise covariance matrices from an empty room scan of MEG which was recorded absent a subject in order to maintain spontaneous neural activity following cortical reconstruction [2].

Subject recordings were registered and resampled onto a common template that resulted in a registered cortical space. The all-by-all phase-locking adjacency matrix was then calculated from each subject's source space data using 8,196 dipoles. Each calculation took approximately 2 hours per 5-minute data file (16.7 million PLV computations). Phase locking adjacency matrices were calculated for each subject and then averaged in the common atlas space to produce a single centrality value at a given frequency prior to clustering.

## 7.4 RESULTS

### 7.4.1 Numerical Simulations

Here we present our results of the numerical simulation studies. We first show the networks identified via our method with respect to the simulated network configuration. Next we look at the effects of the point-spread function on our ability to resolve networks at successively smaller distances between the two simulated point sources.

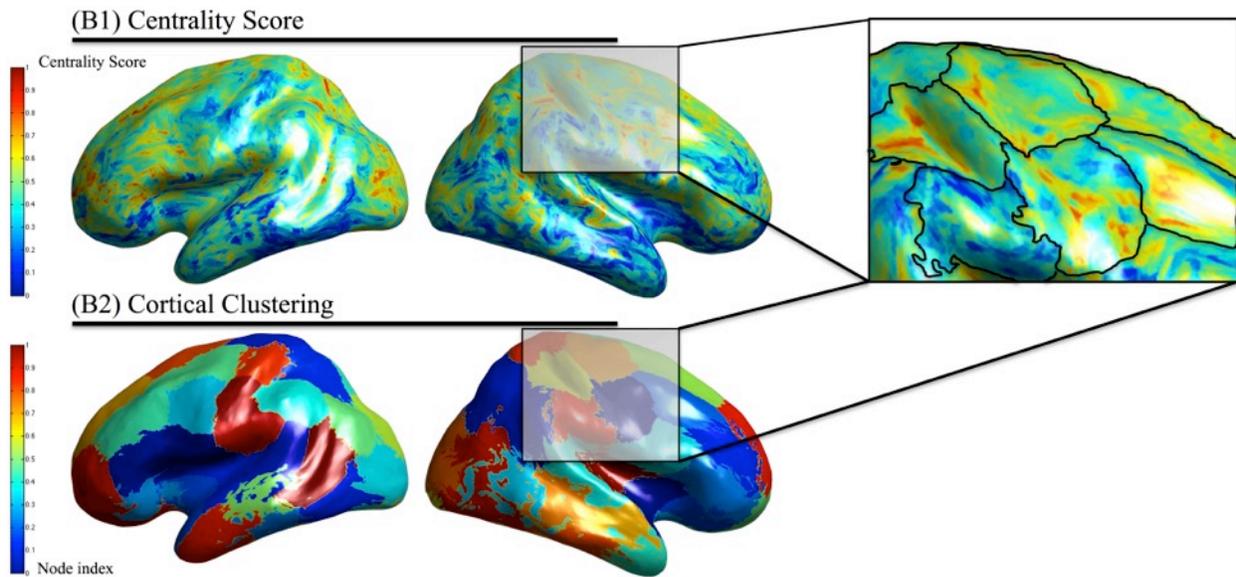
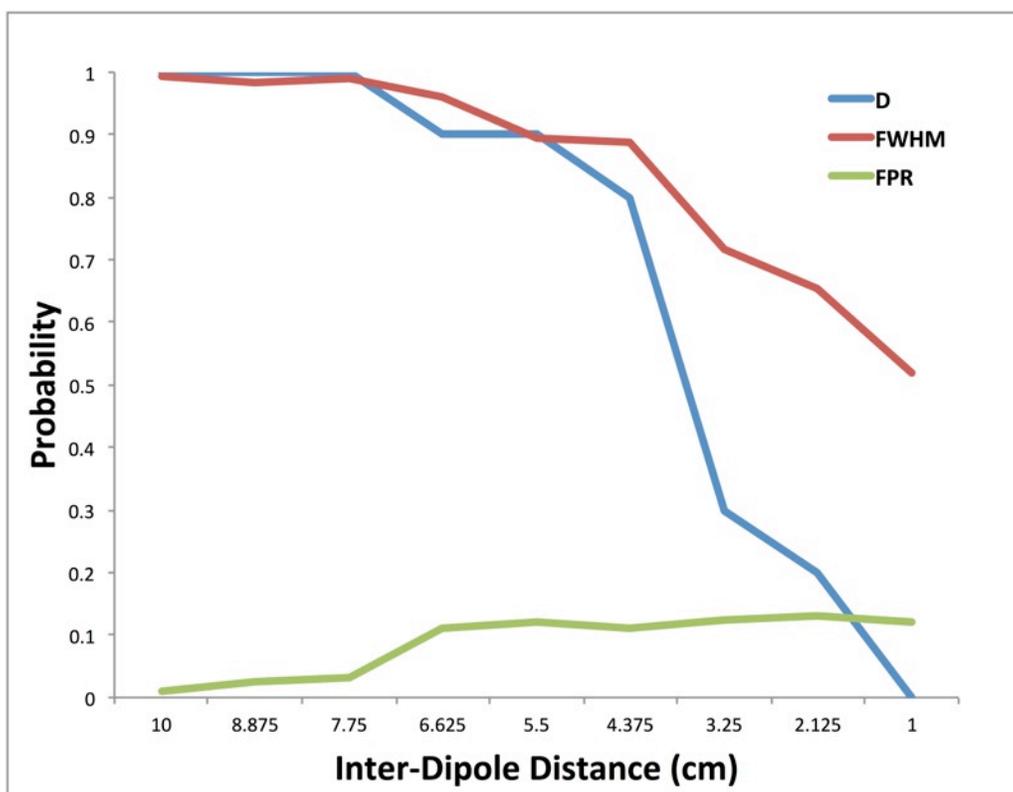


Figure 7-4 Identification of functionally defined regions in subject data

The eigenvector centrality at each cortical location averaged across multiple subjects (Figure 7-4) is used as input to the clustering algorithm that identified regions with similar centrality scores that are nearby on the cortical surface (below). The overlaid regions are depicted on the centrality scores (right).

Functionally defined regions are determined according to the membership of dipole locations to a specific cluster. Each region is spatially contiguous along the cortical surface and possesses dipoles that have similar centrality scores. In the top part of Figure 7-4, the centrality score is displayed on the cortical surface in which higher values (red) indicate a greater importance of that dipole within the phase locking graph. The lower portion of Figure 7-4 shows the result of clustering on the centrality scores to determine the functionally defined regions where surface colors are used to represent difference clusters. Finally, Figure 7-4 shows an overlay of the centrality scores with the clustering method showing similar centrality-value surface dipoles within the functionally defined regions.



**Figure 7-5 TPR of FWHM and Dipole containing clusters.**

The probability of the cluster containing the dipole being significantly connected is plotted with the probability that at least one cluster within the FWHM was significantly connected (Figure 7-5). These are plotted as a function of the distance between the two original dipoles used in generating the simulation. In addition, the false positive rate is shown for clusters significantly connecting outside the FWHM

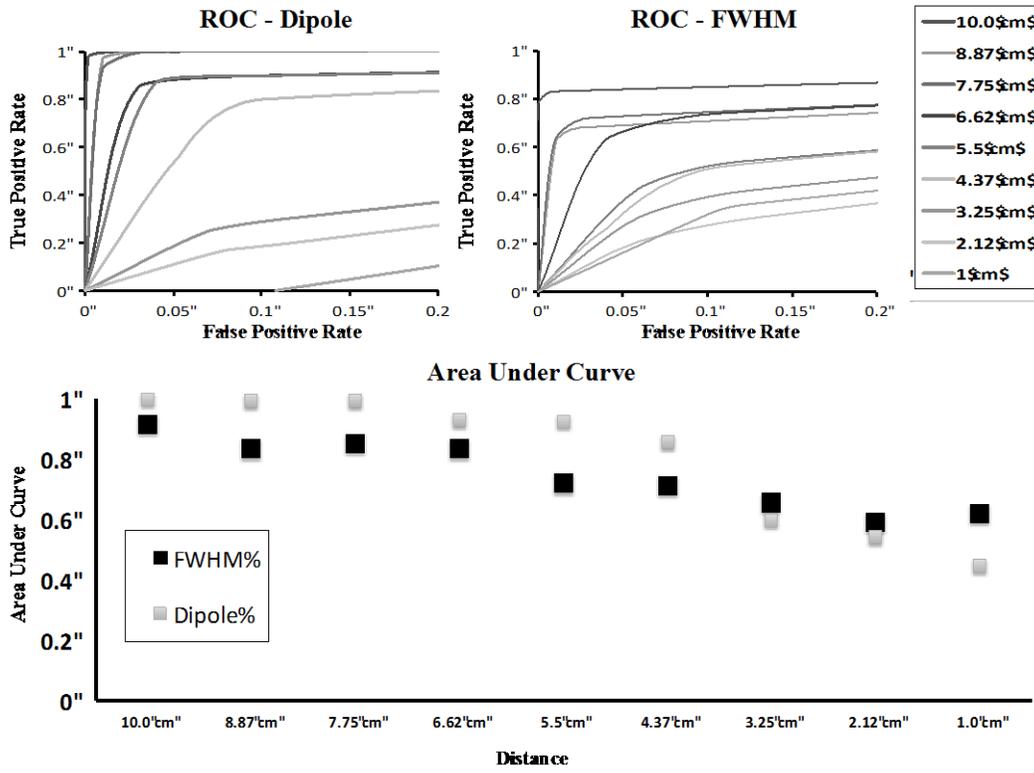


Figure 7-6 ROC curves for brain simulation with area under curve

In Figure 7-6, ROC curves are shown for decreasing inter-dipole distance in the brain simulations for both the cluster containing the originally activated dipole as well as the FWHM region of clusters. As the distance decreases, the number of true positives falls more quickly with only the dipole, but the FWHM maintains resolution at smaller distances

#### **7.4.2 Performance of data-driven methods**

The results of our numerical simulations are shown in Figure 7-5. Also shown is the probability of the cluster that contains the first simulated dipole point source to be connected to the accompanying cluster at the second simulated dipole point source as a function of the distance between the two dipoles. This plot is the true positive rate (TPR) of the dipole-containing clustering. Figure 7-5 also presents the probability that at least one of the FWHM clusters is positively connected to another FWHM cluster at a  $\alpha = 0.05$  level, computed using a cumulative distribution function. At large separation distances both estimation methods are able to identify the connection between the two point sources ranging from a 10 cm to 7.75 cm inter-dipole distance. At inter-dipole distances less than 7.75 cm, there is a divergence between the sensitivity measurements such that at 2.25 cm inter-dipole distance the probability of at least one cluster in the FWHM containing a positive connection is 0.65 while the probability that the cluster containing the original dipole is 0.2. The FWHM maintains a higher sensitivity measurement as the inter-dipole distance decreases because of the decreased spatial sensitivity of the FWHM.

Also in Figure 7-5, we show the false positive rate (FPR) for the combination of the dipole-containing cluster as well as the FWHM clusters. False positives were calculated from those clusters within the FWHM as well as the cluster containing the dipole and similarly for the true negative where all clusters outside the FWHM should be unconnected. There is an increase in the FPR as the inter-dipole distance decreases.

In Figure 7-6 we show the ROC curve for both the dipole only condition as well as the FWHM condition. In the dipole condition, as the two generators move closer, the ability of the method to resolve the connection drops as indicated in Figure 7-5, but the ROC curve shows the rate at which this is occurring. For the FWHM, the ROC curve is moving downwards, but less quickly than the dipole-only. A plot of the difference between the Euclidean distance between the FWHM centroid location and the originally simulated dipole is shown below. As the distance between the dipoles decreases, the FWHM clusters tend to be located non-uniformly around the original dipole location.

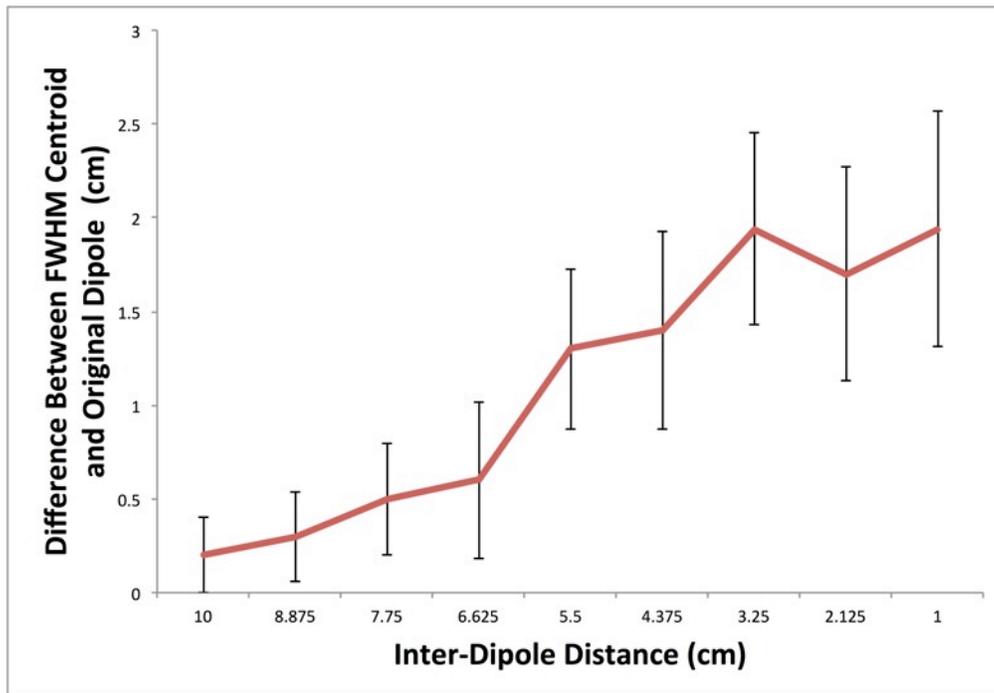


Figure 7-7 FWHM centroid location in simulation as a function of inter-dipole distance

### 7.4.3 Characterization of errors in cluster spatial location

In order to characterize errors introduced at decreasing inter-dipole distances we calculated the offset between the simulated dipole point source and the center of mass of all FWHM cluster centroids that contained at least one significant connection. The result of this analysis is shown in Figure 7-7. At the largest inter-dipole distance of 10 cm, there is an average 0.2 cm (standard deviation: 0.21 cm) discrepancy between the centroid locations indicating a good agreement between the centroid of the FWHM clusters and the dipole location indicating a more uniform grouping of significant clusters surrounding the original dipole location. At decreasing distances, there is a divergence in the two centroids as significant FWHM clusters are grouped more distally from the other dipole resulting in a larger difference between the center of mass of the

FWHM clusters and the dipole location. This effect is most pronounced at the smallest inter-dipole distance of 1 cm in which the centroid discrepancy is 1.94 cm (standard deviation 0.63 cm).

In agreement with Ghuman et al. [2], these simulations show that when the distance between two points decreases, the ability to resolve the two points independently decreases. This can be seen as a direct result of the decreased resolution induced by the inverse solution. However, the inclusion of the FWHM radii and the clusters contained therein allows for an increased resolution at smaller inter-dipole distances as seen in Figure 7-5 as a shift to the right compared to the cluster containing the dipole.

#### **7.4.4 PLV characteristics**

Our previous simulation used a single sinusoidal generator to which additional noise was added to provide realistic SNR and PLV values based upon realistic data. However, to better understand how variations in the underlying signal affect the outcome of our method, we used a variety of perturbations on the original simulation methodology that included SNR modifications, low frequency signal modulations, phase variances as a function of time as well as our ability to resolve off-frequency sinusoids.

After generating our grid of simulation data (Figure 7-8), we applied our clustering methodology to identify significant clusters of activity. We considered the 10x10 grid that contains the generators (the simulated brain activity) to be a true positive if it connected to the cluster containing the other generator. The results of our simulation are shown in Figure 7-9. The first curve displays modifications to the amplitude of the sinusoid relative to the background

noise and therefore the SNR as a multiple of the real data SNR from above. As the SNR increased, the TPR increased due to the higher resolution, as expected, with decaying SNR the ability of our method to accurately identify the correct clusters subsequently decreases. However, for low-frequency modulations, the ability of our method to identify significant clusters was not affected by the addition of a low-frequency modulator. Because of our use of the wavelet methods for identifying instantaneous frequencies, the addition of a low-frequency modulator on the original signal does not have much of an effect on our outcomes.

We also manipulated the phase variance by explicitly using an instantaneous phase component of the original sinusoid that contains a distribution with mean zero and uniformly distributed about a given band. This has a pronounced effect on graph outcomes as we can see from the ROC curve. This is a direct result of both instantaneous phases changing, and therefore making phase locking more difficult, but it is also a result of changing frequency components. As the instantaneous phase increases in volatility, the ability to uniquely resolve the original frequency becomes more difficult. Finally, we investigated unequal frequencies. At a small off-frequency (20 Hz compared to the reference 25 Hz) we are still able to resolve the correct cluster connection as a result of the wavelet time-frequency resolution. That is, small frequency offsets cannot be uniquely distinguished as a result of the characteristics of time-frequency resolution. However, at large frequency offsets the identification of positive clusters drops precipitously. This is again owing to the wavelet method which only resonates over a given frequency band (with a Gaussian envelope) and as a result, off-frequencies do not produce significant connections.

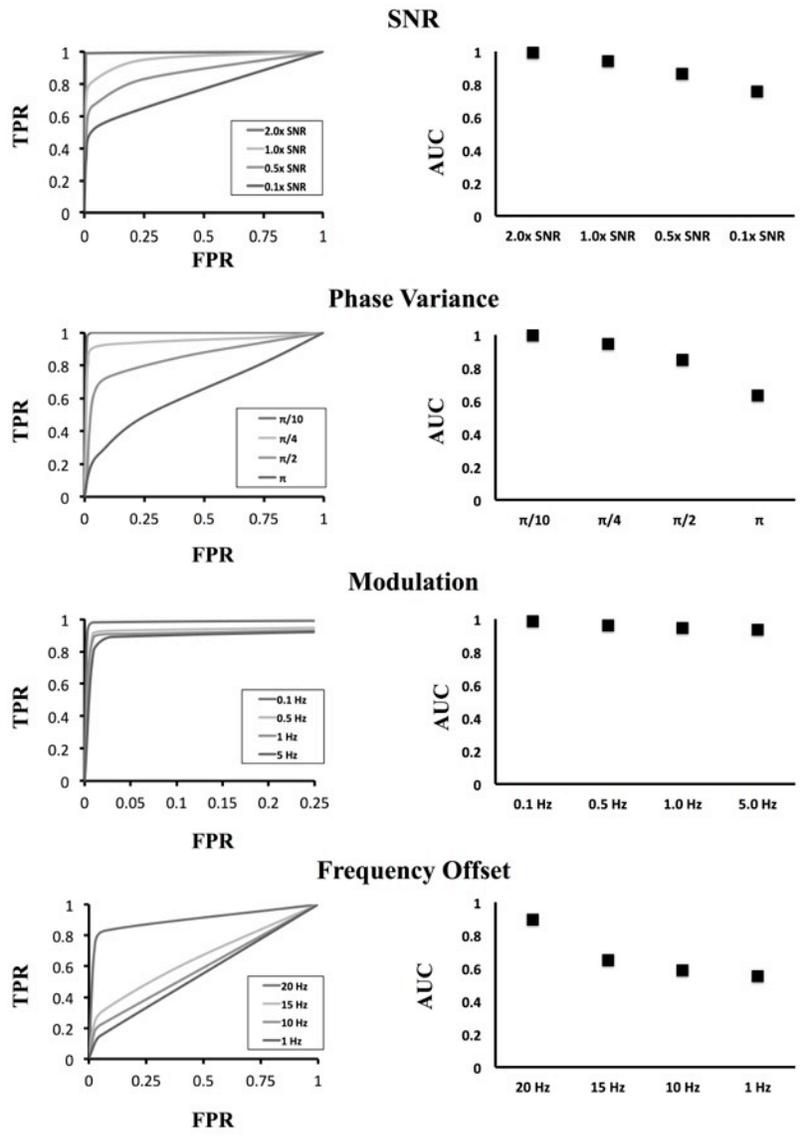


Figure 7-8 ROC Curves of simulation.

#### **7.4.5 Point spread function limitation**

One reason for a discrepancy in centroids locations arises because as the inter-dipole distance decreases, the FWHM radius around each point source begins to have overlapping clusters. That is, both dipoles' FWHM can contain the same cluster. As a result, our statistical test will not find a difference between a population and itself and therefore these clusters are precluded from being significantly connected. As a result of this overlap, the significantly connected clusters will tend to group around the distal portions of the FWHM radii of the two dipoles. We see the effect of this distal grouping in Figure 7-7 as a large discrepancy in the center of mass of the FWHM positive centroids.

The MEG reconstruction method limits the ability to resolve small inter-dipole point sources. However, our numerical simulations show that we are able to obtain good estimates of the reconstructed cortical activity despite this poor resolution. We have also shown that our method produces a small FPR. Finally, the FPR is low through a large number of inter-dipole distances that further validates our statistical methodology and improves this methods utility in neuroscientific investigations. Given our numerical simulation results, we next applied our method to experimental data.

#### **7.4.6 Evaluation of statistical methods**

We evaluated our statistical method by verifying the estimated critical value after calculating the false discovery rate (FDR). Using two empty room datasets as input, we applied our method to calculate the phase locking network using one dataset as the null hypothesis. We used our chosen critical value of  $\alpha = 0.05$  and subsequently calculated the number of positive

connections and repeated this evaluation for 10 pairs of empty room data. From this process we estimated the FDR for our method to have an average FDR of 0.043 with a standard deviation of 0.011.

## **7.5 EXPERIMENTAL RESULTS**

All data were visually inspected prior to processing for anomalies. Following preprocessing of the sensor space data, the transformed data were again visually inspected to ensure no abnormalities arose during preprocessing. The network was analyzed at the alpha (8-13 Hz) frequency band to demonstrate the efficacy of our method by averaging PLV activity estimated at integer values within our range.

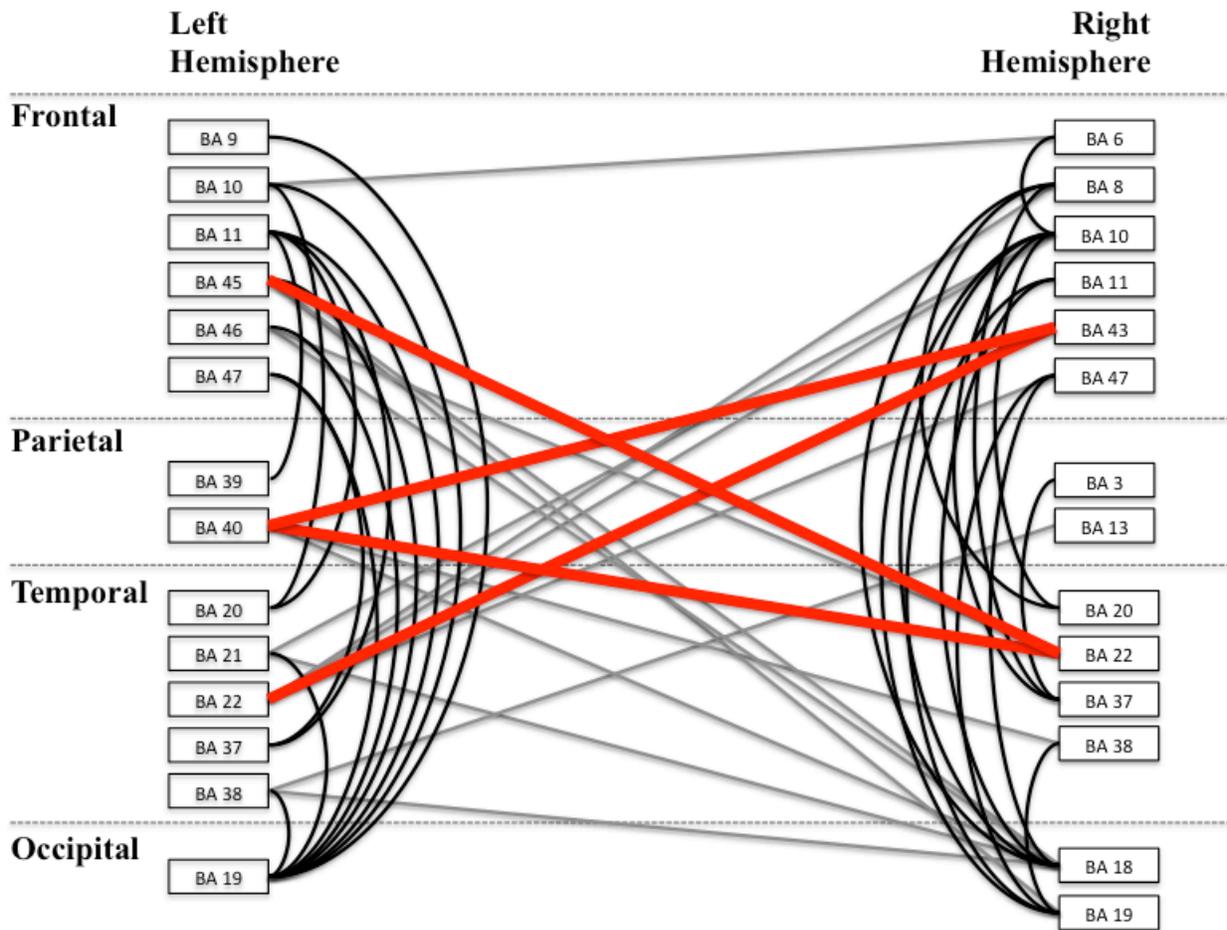
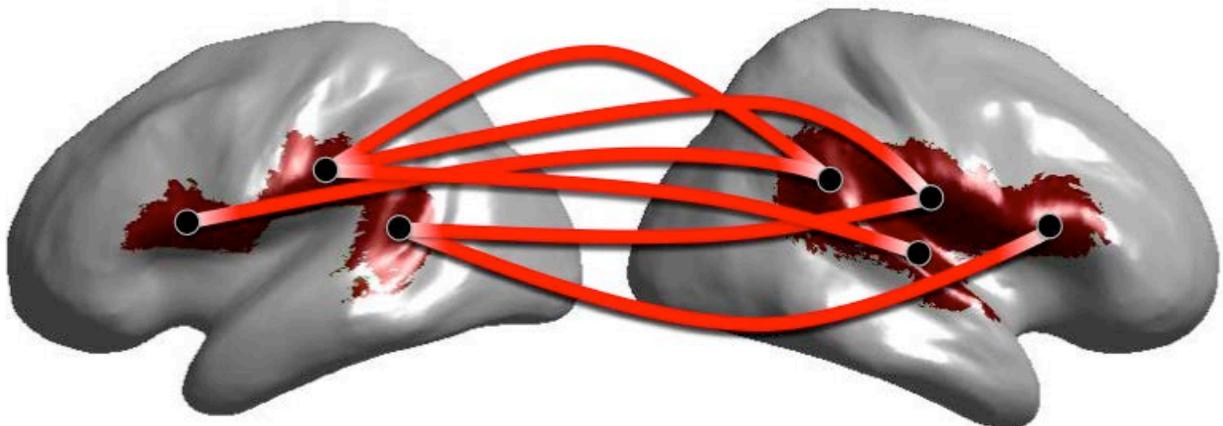


Figure 7-9 FWHM centroid location in simulation as a function of inter-dipole distance

The alpha band phase locking network is shown Figure 7-9. Labeled regions correspond to clusters that were significantly connected to at least one other region. Lines between regions indicate significant phase locking between those regions. The nearest Brodmann area to each functionally defined region was used for labeling. The graph presented is a depiction of the alpha band frequency phase locking connections. The difference between the Euclidean distance between the FWHM centroid location and the originally simulated dipole are plotted. As the distance between the dipoles decreases, the FWHM clusters tend to be located non-uniformly around the original dipole location.



**Figure 7-10 Graph of alpha band network clusters over the arcuate fasciculus**

Figure 7-10 shows a subset of full alpha band network depicting cluster centroids (black dots) overlaid on cluster locations on the cortical surface (red surface). Red lines between centroid locations indicate significant network connections. Cortical regions corresponding to the significant clusters are depicted on the cortical surface. Connections are shown between clusters

centroids as lines between circular discs that are displayed at the centroid of each region. Our method reveals phase locked connections between contralateral regions of the arcuate fasciculus. Previous studies have shown the arcuate fasciculus plays a role in language [135] and furthermore, that functional connections exist between these cortical locations and specifically at the alpha band [2].

## 8.0 DISCUSSION

Whole brain functional connectivity on healthy normal subjects has revealed complex network interactions at multiple physiological frequency bands. Limitations arising from the inverse solution that leads to degradation in resolution, and results in a decreased ability to detect spatially nearby functionally connected regions is common in other studies of MEG functional connectivity [2]. To mitigate these effects and provide a tool that neuroscientists can utilize to identify complex cortical network interactions we have introduced a method that incorporates dipole level centrality metrics obtained from graph theory to identify regions with similar connectivity patterns. From these regions we were then able to identify significant connections utilizing a statistical permutation test.

Clustering of graph vertex centrality scores is a novel method for identifying functionally defined regions on the cortical surface. Our choice of spherical clustering methods produces regions that are spatially contiguous without disjointed surfaces and are therefore more easily interpretable. Functionally defined regions have been used previously in establishing cortical locations with overlapping functional behavior [28, 126, 127]. In general, these methods provide a data-driven method for assessing membership of individual cortical locations and assigning groups to regions. In contrast, anatomically defined regions that are defined based upon the physical connectivity of neural populations [110]. The size of the dipole level adjacency matrix necessitated the choice of a centrality measure that is computable in reasonable time upon a

roughly 8,000 square matrix. Specifically, we used eigenvector centrality as our measure of similarity between vertex locations as it is a widely used metric within graph theory and it was computationally tractable given the dimensionality of our adjacency matrix.

Our use of permutation tests on the region-by-region statistical analysis provided a test that explicitly accounts for the underlying distributions of phase locking values between regions. This is an important property of permutation tests – that they are non-parametric and do not rely upon the underlying distribution of the data conforming to any specific parametric distribution – and is the primary reason we choose to use the permutation test in our analysis. This is especially important when testing the highly dependent dipole level activity that results from cross talk arising from the MEG reconstruction method. Parametric models of dipole level activity are susceptible to deviations in the expected type I error as a result of violations in the assumptions in the underlying distribution. Importantly, we demonstrated that our method has a low FDR in the simulations producing a small false positive population.

Application of whole brain phase locking estimates to subject data at the alpha band revealed complex network interactions across many areas of the cortical surface. Importantly, these networks arise without a hypothesis driven expectation of the cortical spatial location of the neurophysiological activity. Furthermore, networks that have been studied in literature are revealed by our whole brain analysis utilizing only these data-driven methods. As such, these results provide powerful evidence for the data-driven establishment of alpha band networks. Connections that were previously unknown provide a rich area of future studies to identify their important integration into whole-brain cortical activity.

## 8.1 LIMITATIONS

Owing to the ill-posed nature of the MEG inverse problem, there is no unique solution to the reconstruction of cortical activity. Here we have used the MNE estimation because it makes few assumptions about the covariance structure of the data. MNE is a Tikhonov regularization that uses a scalar identity matrix for the covariance structure. In the context of resting state functional connectivity, the covariance structure is important to understanding the temporal relationships between cortical locations. Alternative methods could provide additional information from resting state data owing to their use of different underlying assumptions. In addition, alternative methods may be more robust to the point spread function. Decreasing the influence of the point spread function on nearby points would allow for more granular assessments of the phase locking between regions. As our simulation results indicated, the point spread function prevents short-distance connections from being significantly detected.

In the present work we have used k-means clustering because it is fast, computationally simple, and provides robust estimates for clusters based upon an input space. Given our data-driven focus, we wished to provide a clustering method that establishes membership of every cortical location to a given cluster. In this way, we were able to compute the whole brain functional connectivity mapping. K-means is known to produce clusters with similar spatial extents, which can be seen in Figure 7-4 owing to its variance-based clustering heuristic. This is especially useful in top-down modeling in which we first assign all points to the cluster and eliminate those clusters that are not significant. In addition, k-means does not provide a robust mechanism for estimating the number of clusters. In the present work we used a dendrogram to identify a reasonable tradeoff between computational complexity (increasing clusters increases the computational time by the square of the number of clusters) and similarity between dipoles

contained within the cluster. Owing to this tradeoff, we have underestimated the total number of clusters. As a result, regions that contained populations of more than one similarity may produce a higher type II error rate.

However, using a bottom-up approach and more complex clustering model one could aggregate related functional locations into significant clusters. Gaussian mixture models have been used in developing these bottom-up models of functionally related cortical locations [136, 137]. In that work, mesostate models aggregate related dipole-level activity into regions of similar activity and provide a bottom-up model generation technique for understanding functional connectivity. Interestingly, mesostate models can incorporate priors into their analysis, and one such prior could be the outcome of our methodology providing the top-down view of the whole-brain network architecture. In addition, these alternative methods for clustering could provide methods to empirically estimate the number of clusters and eliminating the need for our dendrogram method for estimating a  $k$  for  $k$ -means.

We used a time-averaged PLV that aggregates each time-series into a single value. It is well known that MEG contains many non-stationarities in the data and that those non-stationarities can be directly investigated using windowing methods [89]. Time averaging over the scan period can only capture processes that are jointly stationary. The use of wavelets in our work results in a methodological framework that is not as susceptible to processes that are non-stationary when estimating the instantaneous phase of each signal independently as compared with traditional Fourier methods. However, PLV is still susceptible to signals that are not jointly stationary (i.e. if each signal changes the underlying properties together we are still able to resolve that relationship even if they are each non-stationary processes). Time windowing would provide a method to investigate this property directly by analyzing smaller windows of time (i.e. over a short duration window the signals are jointly stationary).

## 8.2 FUTURE DIRECTIONS

We have presented a methodological framework for assessing whole brain functional connectivity networks. Our method is unique in that it attempts to provide simultaneous assessments of the underlying cortical network architecture at specified frequency band without requiring knowledge of the expected network connections. This data-driven approach to whole brain functional connectivity provides a model-free view of cortical networks and provides a starting point for future investigations of individual network connections. We believe that our method is a powerful tool for identifying previously undiscovered networks. However, our method provides a very high-level understanding of whole-brain networks. The combination of generative models of dipole-level activity and our top-down modeling approach would provide researcher the methodological framework to investigate functional connectivity patterns over a large range of scales.

The use of graph theory for analysis of cortical networks provides a toolset for assessing various aspects of brain health and quantitative metrics for characterizing those networks. Future studies could incorporate these graph theory quantification metrics upon revealed networks. For instance, they could be used to assess differences between frequency bands or subject populations. These comparisons would then be capable of providing neuroscientific insights into cognitive processes in a quantitative manner that could ultimately lead to an improved understanding of cognitive disease models.

## **APPENDIX A**

### **DEMO CODE**

#### **A.1 WHOLE BRAIN PHASE LOCKING**

##### **A.1.1 Overview**

Here we describe the process of transforming an input consisting of a matrix of time series recordings at various locations into a matrix of phase locking values between each pairwise location. The input matrix is of size  $n$  by  $t$ , where  $n$  is the number of locations and  $t$  is the length of the time series recordings.

Legion, as described in chapter 4, requires that we proceed with the calculation in a few steps. First, we separate the input data into blocks and write them to disk. Next, we launch the job to calculate each pair of blocks. After each sub-task has completed, we reintegrate the answers into a single matrix. This is a form of scatter-gather processing.

All software is provided as an accompaniment of the Legion software.

## A.2 ALL-BY-ALL PLV EXECUTION

We begin by writing the data to an output directory. To write data into blocks we use the function:

```
legion.stream.Util.write_split_data( Data, Output_path, NumSplits );
```

Where `Data` is the input matrix of `n` by `t`, `Output_path` is a string to a directory to write the file blocks to and `NumSplits` is an integer with the number of files to write to. Each file will be a MAT file labeled as `split_<idx>`, where `idx` represents the split index in the original input data. Paths are relative to the `Grid` path (`legion.stream.Grid`) which defaults to `/synapse/logs/<username>`.

Next, we create a `Legion` instance that we can configure before submitting the job to the cluster.

```
master = legion.stream.Master(2);
master.setInputPath( input_path );
master.setOutputPath( output_path );
```

We create an object, `master`, which we initialize with a `2` to indicate the pair-wise calculation scheme. We then set the master's properties corresponding to the input and output paths for processing. The `input_path` should be similar to the output path of the previous block writing path. The output path is the location for each thread to write its answer. Again, note that paths are relative to the `Grid` path.

Next we must specify various kernels that are used for preprocessing of blocks and the actual block processing code. These allow for maximum variability in processing streams. For PLV analysis, the following is the optimal processing scheme:

```

% Block Reading Kernel
read_kern = pitt.exp.plv.Kernels.read_and_complex_preprocess_kernel( Fs,
Freqs );
master.setReadKernel( read_kern );

% PLV Processing Kernel
preproc_kern= pitt.exp.plv.Kernels.plvcomplexkernel(Fs, Freqs);
master.setThreadKernel( preproc_kern );

% Block Saving Kernel
save_kern = pitt.exp.plv.Kernels.save_by_frequency_kernel_v2();
master.setSaveKernel( save_kern );

```

Three kernels are specified. The first is the read kernel. This kernel is executed on a block of data (blocks as defined in the splitting operation). It accepts a number of channels by number of time points data block. Here we use the `read_and_complex_preprocess_kernel` which preprocesses the block using the wavelet transform into complex signals. `Fs` is the sampling frequency of the time series and `Freqs` is a vector of Integer frequency values upon which to evaluate the wavelet. At current settings, the optimal number of frequencies to calculate in a single processing is four.

The next kernel specifies the process to use on each pair of channels. Here we use a kernel which accepts already complex inputs before computing the PLV calculation. Finally, we set the save kernel. This kernel, `save_by_frequency_kernel_v2`, writes each frequency specified in the `Freqs` variables into a unique folder at the `output_path` named as `freq_<frequency>.mat`

Lastly, we specify the working directory used to maintain the QSUB output logs and intermediate files used during processing. This directory location must be unique. It is also relative to the Grid path as above. And lastly we specify the number of processors to use in processing, NumProcs.

```
master.setGridWorkingDirectory( grid_job );  
master.setNumThreads( NumProcs );
```

Note that the number of threads can be set to higher or lower than the available number of physical processors. If the value is higher, Legion will queue up jobs and will only execute them after the initial batch has finished (this is inefficient). The optimal solution is to specify exactly the number of available cores or fewer. In this case, Legion will execute MATLAB jobs on that number of processors which will be active until all block pairs have finished executing.

Finally, we submit the job to the QSUB:

```
master.submit_job();
```

Which will print to screen the QSUB commands used to execute the work. When all the jobs have completed (this must be checked manually, use “qstat -f” on the command line), each pair of jobs will have been written to their frequency directories at the output path specified. To reintegrate these split files into a single matrix the following must be run for each frequency directory:

```
plv_array = legion.utils.reintegrate( output_dir );
```

This will produce an array, plv\_array, which is a square matrix with each dimension equal to the number of channels. Only those values in the lower left triangle will have been computed. One of these matrices must be created for each frequency of interest.

### A.3 ARBITRARY SCRIPTING

From the above example, one can create an arbitrary processing pipeline that operates on two input series of type, [channel x data]. This can be accomplished by changing the kernel operation performed. Reasonable default values are given for both the read and save kernels, such that it is typically not necessary to override them.

Chapter 4 gave the API for Kernels but here we provide an example of a simplistic kernel. We wish to compute the mean difference between each pairwise combination of input signal. To accomplish this, we use the following kernel:

```
ker = legion.Kernel()  
f = @(x) mean(x(1,:)) - mean(x(2,:))  
ker.add( @f, 'X' )
```

We first build a new kernel object, `ker`. We define an anonymous function which accepts a matrix of size 2 x data and compute the mean difference between each row of the two data sets. We then add this function as a processing handle on the kernel object using the `add()` function. The 'X' argument is provided so they system is aware of where to pass arguments. This new kernel function can then be added in a method similar to above as:

```
master.setThreadKernel( ker );
```

Kernels can also be used as arbitrary processing pipelines that can chain a series of functions passing the outputs of earlier function into later arguments. This can be shown with the following example:

```
ker = legion.Kernel()
ker.add( @mean, 'X', 2 )
ker.add( @sum, 'X' )
ker.initialize( randn( 10, 4 ) )
output = ker.output()
```

This sequence will perform an operation equivalent to the following:

```
output = sum( mean( randn( 10,4 ), 2 ) )
```

In this way, kernels are like anonymous functions with built-in pipelining. The ability to specific arbitrary arguments (e.g. above the sum in the second dimension) allows extensibility. And the object oriented nature provides a mechanism for encapsulating state information in kernels for passing to distributed systems.

## A.4 POST PROCESSING CODE

To perform the post processing, we must first obtain the subject specific meta data of the brain spaces. The following script will obtain the Opt subject data:

```
pitt.cerebro.Scripts.scripts()
```

Which contains the following code:

```
opt001 = pitt.cerebro.Reg.genUserData( 'Opt001' );
    opt001 = pitt.cerebro.Reg.appendObservedData( opt001, 'rest2',
'~/data/r21dataNM/Opt001_resting2_ds4_filt_1_50-ave-Opt001-oct-6-src-fwd.fif' );
    %opt001 = pitt.cerebro.Reg.appendObservedData( opt001, 'rest2',
'~/data/r21dataNM/Opt001_resting1_projon_ds4_filt_0_50_tSSS-ave-Opt001-oct-6-src-meg-inv.fif' );
    %opt001 = pitt.cerebro.Reg.appendObservedData( opt001, 'rest2', '~/fwd.fif' );
    opt001 = pitt.cerebro.Reg.appendObservedData( opt001, 'empty',
'~/data/r21dataNM/Opt001_resting1_projon_ds4_filt_0_50-ave-Opt001-oct-6-src-fwd.fif' );
    opt001 = pitt.cerebro.Reg.smoothLoResToHiResInSphere( opt001, 'empty' );

diffL = mean(opt001.empty.surface.L.v) - mean(optColin.pial.surface.L.v);
```

```

diffR = mean(opt001.empty.surface.R.v) - mean(optColin.pial.surface.R.v);

vL = opt001.rest2.surface.L.v - repmat(diffL, size( opt001.empty.surface.L.v,1 ),1
);
vR = opt001.rest2.surface.R.v - repmat(diffR, size( opt001.empty.surface.R.v,1 ),1
);

kL = dsearchn( vL, optColin.pial.surface.L.v );
kR = dsearchn( vR, optColin.pial.surface.R.v );

opt001.empty.reg.colin.L = kL;
opt001.empty.reg.colin.R = kR;

% -----
% Opt052
% -----
opt052 = pitt.cerebro.Reg.genUserData( 'Opt052' );
opt052 = pitt.cerebro.Reg.appendObservedData( opt052, 'rest2',
'~/data/r21dataNM/Opt052_resting2_projon_ds4_filt_0_50_tSSS-ave-Opt052-oct-6-src-fwd.fif' );
opt052 = pitt.cerebro.Reg.smoothLoResToHiResInSphere( opt052, 'rest2' );
opt052 = pitt.cerebro.Reg.regiserLoResToHiResInNewSpace( opt052, optColin, 'rest2',
'colin' );

% -----
% Opt060
% -----
opt060 = pitt.cerebro.Reg.genUserData( 'Opt060' );
opt060 = pitt.cerebro.Reg.appendObservedData( opt060, 'rest2',
'~/data/r21dataNM/Opt060_resting2_projon_ds4_filt_0_50_tSSS-ave-Opt060-oct-6-src-fwd.fif' );
opt060 = pitt.cerebro.Reg.smoothLoResToHiResInSphere( opt060, 'rest2' );
opt060 = pitt.cerebro.Reg.regiserLoResToHiResInNewSpace( opt060, optColin, 'rest2',
'colin' );

% -----
% Opt062
% -----
opt062 = pitt.cerebro.Reg.genUserData( 'Opt062' );
opt062 = pitt.cerebro.Reg.appendObservedData( opt062, 'rest2',
'~/data/r21dataNM/Opt062_resting2_projon_ds4_filt_0_50_tSSS-ave-Opt062-oct-6-src-fwd.fif' );
opt062 = pitt.cerebro.Reg.smoothLoResToHiResInSphere( opt062, 'rest2' );
opt062 = pitt.cerebro.Reg.regiserLoResToHiResInNewSpace( opt062, optColin, 'rest2',
'colin' );

% -----
% Opt065
% -----
opt065 = pitt.cerebro.Reg.genUserData( 'Opt065' );
opt065 = pitt.cerebro.Reg.appendObservedData( opt065, 'rest2',
'~/data/r21dataNM/Opt065_resting2_projon_ds4_filt_0_50_tSSS-ave-Opt065-oct-6-src-fwd.fif' );
opt065 = pitt.cerebro.Reg.smoothLoResToHiResInSphere( opt065, 'rest2' );
opt065 = pitt.cerebro.Reg.regiserLoResToHiResInNewSpace( opt065, optColin, 'rest2',
'colin' );

% -----
% Opt067
% -----
opt067 = pitt.cerebro.Reg.genUserData( 'Opt067' );
opt067 = pitt.cerebro.Reg.appendObservedData( opt067, 'rest2',
'~/data/r21dataNM/Opt067_resting2_projon_ds4_filt_0_50_tSSS-ave-Opt067-oct-6-src-fwd.fif' );
opt067 = pitt.cerebro.Reg.smoothLoResToHiResInSphere( opt067, 'rest2' );
opt067 = pitt.cerebro.Reg.regiserLoResToHiResInNewSpace( opt067, optColin, 'rest2',

```

```
'colin' );
```

```
opts = struct;  
opts.optColin = optColin;  
opts.opt001 = opt001;  
opts.opt052 = opt052;  
opts.opt060 = opt060;  
opts.opt062 = opt062;  
opts.opt065 = opt065;  
opts.opt067 = opt067;
```

This will create an `opts` structure containing the relevant meta data including the structural, spherical and registered spherical spaces. In addition, it will resample the spaces such that transformation between subjects can occur via the included Colin brain space data.

Next, we perform the calculation on the frequency-specific data:

```
X = pitt.cerebro.Scripts.processBand( band, opts )
```

which will execute the following code:

```
x          = struct;
x.band    = band;

% -----
% Freq Data
% -----
fprintf( 'Loading \n' );
x.freqopt001r = pitt.cerebro.DataLoader.loadOptRest(
['~/data/all/opt001/empty_done/freq_',band] );
x.freqopt052r2 = pitt.cerebro.DataLoader.loadOptRest(
['~/data/all/opt052/rest2_done/freq_',band] );
x.freqopt060r2 = pitt.cerebro.DataLoader.loadOptRest(
['~/data/all/opt060/rest2_done/freq_',band] );
x.freqopt062r2 = pitt.cerebro.DataLoader.loadOptRest(
['~/data/all/opt062/rest2_done/freq_',band] );
x.freqopt065r2 = pitt.cerebro.DataLoader.loadOptRest(
['~/data/all/opt065/rest2_done/freq_',band] );
x.freqopt067r2 = pitt.cerebro.DataLoader.loadOptRest(
['~/data/all/opt067/rest2_done/freq_',band] );

% -----
% EVC Calculation in Self-Space
% -----
fprintf( 'EVC' );
x.evcopt052r2 = pitt.cerebro.EVC.eigenVectorCentrality( x.freqopt052r2 );
fprintf( '...done' );
x.evcopt060r2 = pitt.cerebro.EVC.eigenVectorCentrality( x.freqopt060r2 );
fprintf( '...done' );
x.evcopt062r2 = pitt.cerebro.EVC.eigenVectorCentrality( x.freqopt062r2 );
fprintf( '...done' );
x.evcopt065r2 = pitt.cerebro.EVC.eigenVectorCentrality( x.freqopt065r2 );
fprintf( '...done' );
x.evcopt067r2 = pitt.cerebro.EVC.eigenVectorCentrality( x.freqopt067r2 );
fprintf( '...done' );
fprintf( '\n' );

% -----
% EVC In Colin Space
% -----
fprintf( 'EVC Colin \n' );
x.evc52 = pitt.cerebro.DispBrain.upsampleToHiResRegisteredDualHemi(
opts.opt052.rest2, x.evcopt052r2 );
x.evc60 = pitt.cerebro.DispBrain.upsampleToHiResRegisteredDualHemi(
opts.opt060.rest2, x.evcopt060r2 );
x.evc62 = pitt.cerebro.DispBrain.upsampleToHiResRegisteredDualHemi(
opts.opt062.rest2, x.evcopt062r2 );
x.evc65 = pitt.cerebro.DispBrain.upsampleToHiResRegisteredDualHemi(
opts.opt065.rest2, x.evcopt065r2 );
x.evc67 = pitt.cerebro.DispBrain.upsampleToHiResRegisteredDualHemi(
opts.opt067.rest2, x.evcopt067r2 );
```

```

fprintf( 'Ave EVC \n' );
mevc = mean([x.evc52;x.evc60;x.evc62;x.evc65;x.evc67]);

x.mevc = mevc;

fprintf( 'Preproc EVC' );
[x.freqopt052r2m, x.evcopt052r2m, x.evc52m] =
pitt.cerebro.Scripts.preprocessWithMeanEVC( opts.opt052, mevc, x.freqopt052r2 ); fprintf(
'...done' );
[x.freqopt060r2m, x.evcopt060r2m, x.evc60m] =
pitt.cerebro.Scripts.preprocessWithMeanEVC( opts.opt060, mevc, x.freqopt060r2 ); fprintf(
'...done' );
[x.freqopt062r2m, x.evcopt062r2m, x.evc62m] =
pitt.cerebro.Scripts.preprocessWithMeanEVC( opts.opt062, mevc, x.freqopt062r2 ); fprintf(
'...done' );
[x.freqopt065r2m, x.evcopt065r2m, x.evc65m] =
pitt.cerebro.Scripts.preprocessWithMeanEVC( opts.opt065, mevc, x.freqopt065r2 ); fprintf(
'...done' );
[x.freqopt067r2m, x.evcopt067r2m, x.evc67m] =
pitt.cerebro.Scripts.preprocessWithMeanEVC( opts.opt067, mevc, x.freqopt067r2 ); fprintf(
'...done' );
fprintf('\n');

x.mfixevc = mean([x.evc52m;x.evc60m;x.evc62m;x.evc65m;x.evc67m]);

% Create clusters
fprintf( 'Calc Idxs \n' );
x.idx = pitt.cerebro.Cluster.kClusterOnSurface( opts.optColin, 'sphere', 40,
x.mfixevc.*100'.^3 );

% Create subject specific clustering on self-space
fprintf( 'LoRes Space \n' );
x.opt1idx = pitt.cerebro.DispBrain.downsampleHiResRegisteredToLowByMode(
opts.opt001, 'empty', x.idx );
x.opt52idx = pitt.cerebro.DispBrain.downsampleHiResRegisteredToLowByMode(
opts.opt052, 'rest2', x.idx );
x.opt60idx = pitt.cerebro.DispBrain.downsampleHiResRegisteredToLowByMode(
opts.opt060, 'rest2', x.idx );
x.opt62idx = pitt.cerebro.DispBrain.downsampleHiResRegisteredToLowByMode(
opts.opt062, 'rest2', x.idx );
x.opt65idx = pitt.cerebro.DispBrain.downsampleHiResRegisteredToLowByMode(
opts.opt065, 'rest2', x.idx );
x.opt67idx = pitt.cerebro.DispBrain.downsampleHiResRegisteredToLowByMode(
opts.opt067, 'rest2', x.idx );

% Obtain surface data corresponding to cluster-cluster freq data
fprintf( 'Orig Data \n' );
x.opt1loCdata = pitt.cerebro.DispBrain.loResPopSampleFromHiResCluster(
x.opt1idx, x.freqopt001er );
x.opt52loCdata = pitt.cerebro.DispBrain.loResPopSampleFromHiResCluster(
x.opt52idx, x.freqopt052r2m );
x.opt60loCdata = pitt.cerebro.DispBrain.loResPopSampleFromHiResCluster(
x.opt60idx, x.freqopt060r2m );
x.opt62loCdata = pitt.cerebro.DispBrain.loResPopSampleFromHiResCluster(
x.opt62idx, x.freqopt062r2m );
x.opt65loCdata = pitt.cerebro.DispBrain.loResPopSampleFromHiResCluster(
x.opt65idx, x.freqopt065r2m );
x.opt67loCdata = pitt.cerebro.DispBrain.loResPopSampleFromHiResCluster(
x.opt67idx, x.freqopt067r2m );

% Generate cell data for rest vs empty conditions
x.erccell = cell(0);
x.erccell(1) = x.opt1loCdata;
x.restcell = cell(0);

x.restcell(1) = x.opt60loCdata;

```

```

x.restcell(2) = x.opt62loCdata;
x.restcell(3) = x.opt65loCdata;
x.restcell(4) = x.opt67loCdata;
x.restcell(5) = x.opt52loCdata;

% Run out permutation testing on cluster-cluster significance
fprintf( 'Stats \n' );
[x.tstats,x.output,x.tpop] = pitt.cerebro.Stats.permute_calc_stats(
x.restcell, x.ercell );

fprintf( 'Done \n' );

```

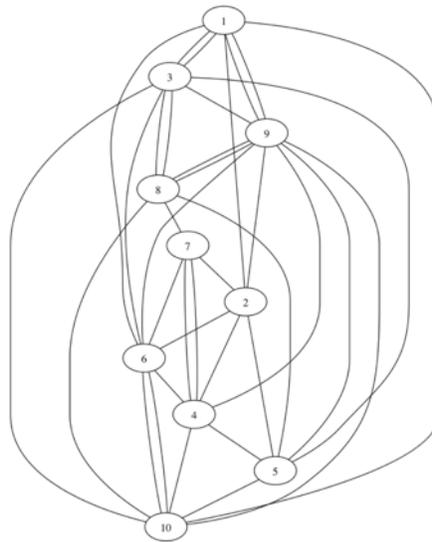
Where band is the string representation of the frequency band data (e.g. “005”) and opts is the structure variable output from the above. The X output is a structure that contains intermediate data regarding the processing as well as information regarding clustering and significant connections in the cluster space.



To generate an undirected graph, use the following code:

```
pitt.viz.graph.Grapher.genNeatoFile( path, adj_matrix, threshold )
```

which given an input of randn(10,10) with threshold of 0.3 will generate a graph similar to the following:



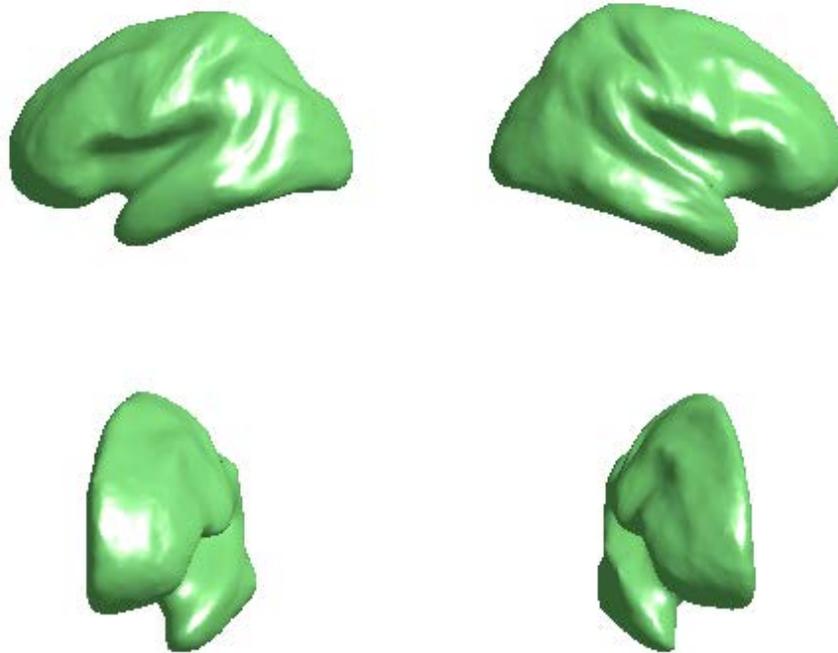
## B.2 BRAIN VISUALIZATION

The following code shows how to display brain visualization in MATLAB. Using the `opts` variable described above in the PLV calculation, we are able to display a variety of brain formats and overlay data upon them without resorting to the `set()` and `get()` handle functions.

To demonstrate, we will first start by showing the Colin brain atlas on the inflated surface:

```
optColin = pitt.cerebro.DispBrain.outputSurfaceImgs( optColin, 'inflated' )
```

The 'inflated' parameter can be values of surface types. These include: inflated, sphere, sphere\_reg, and pial. The above command will produce an image similar to the following:



which shows the same brain in four images. We can then display data overlaid upon the brain using:

```
pitt.cerebro.DispBrain.displayRawDataOntoSurfaceImgs( optColin, 'inflated',  
v );
```

where  $v$  is a vector the size of the total number of nodes on both hemispheres. Each value will represent a display value at that vertex for coloring purposes.

## BIBLIOGRAPHY

1. Friston, K.J., *Functional and effective connectivity: a review.*, in *Brain Connect*2011. p. 13-36.
2. Ghuman, A.S., J.R. McDaniel, and A. Martin, *A wavelet-based method for measuring the oscillatory dynamics of resting-state functional connectivity in MEG*, in *Neuroimage*2011. p. 69-77.
3. Stam, C.J., et al., *Graph theoretical analysis of magnetoencephalographic functional connectivity in Alzheimer's disease*, in *Brain*2009. p. 213-24.
4. Rubinov, M. and O. Sporns, *Complex network measures of brain connectivity: uses and interpretations*, in *Neuroimage*2010. p. 1059-69.
5. Biswal, B., et al., *Functional connectivity in the motor cortex of resting human brain using echo-planar MRI*. *Magn Reson Med*, 1995. **34**(4): p. 537-41.
6. Engel, J., et al., *Connectomics and epilepsy*. *Curr Opin Neurol*, 2013. **26**(2): p. 186-94.
7. Sporns, O., *Networks of The Brain*. MIT Press, 2010.
8. Sporns, O., et al., *Organization, development and function of complex brain networks*. *Trends Cogn Sci*, 2004. **8**(9): p. 418-25.
9. Allison, J.D., et al., *Functional MRI cerebral activation and deactivation during finger movement*. *Neurology*, 2000. **54**(1): p. 135-42.
10. Samuel, M., et al., *Exploring the temporal nature of hemodynamic responses of cortical motor areas using functional MRI*. *Neurology*, 1998. **51**(6): p. 1567-75.
11. van der Kruijs, S.J., et al., *Neurophysiological correlates of dissociative symptoms*. *J Neurol Neurosurg Psychiatry*, 2012.
12. Niedworok, C.J., et al., *Charting monosynaptic connectivity maps by two-color light-sheet fluorescence microscopy*. *Cell Rep*, 2012. **2**(5): p. 1375-86.

13. Collin, G., et al., *Structural and Functional Aspects Relating to Cost and Benefit of Rich Club Organization in the Human Cerebral Cortex*. Cereb Cortex, 2013.
14. Owen, J.P., et al., *The structural connectome of the human brain in agenesis of the corpus callosum*. Neuroimage, 2013. **70**: p. 340-55.
15. Sporns, O., *The human connectome: Origins and challenges*. Neuroimage, 2013.
16. White, J.G., et al., *The structure of the nervous system of the nematode Caenorhabditis elegans*. Philos Trans R Soc Lond B Biol Sci, 1986. **314**(1165): p. 1-340.
17. Dyrby, T.B., et al., *Validation of in vitro probabilistic tractography*. Neuroimage, 2007. **37**(4): p. 1267-77.
18. Parker, G.J., et al., *Initial demonstration of in vivo tracing of axonal projections in the macaque brain and comparison with the human brain using diffusion tensor imaging and fast marching tractography*. Neuroimage, 2002. **15**(4): p. 797-809.
19. Friston, K.J., et al., *Functional connectivity: the principal-component analysis of large (PET) data sets*. J Cereb Blood Flow Metab, 1993. **13**(1): p. 5-14.
20. Bressler, S.L. and A.K. Seth, *Wiener-Granger causality: a well established methodology*, in *Neuroimage*2011. p. 323-9.
21. Baccalá, L.A. and K. Sameshima, *Partial directed coherence: a new concept in neural structure determination*, in *Biol Cybern*2001. p. 463-74.
22. Sommerlade, L., et al., *Inference of Granger causal time-dependent influences in noisy multivariate time series.*, in *J Neurosci Methods*2012. p. 173-85.
23. Tsiaras, V., et al., *Extracting biomarkers of autism from MEG resting-state functional connectivity networks*, in *Comput Biol Med*2011. p. 1166-77.
24. Hesse, W., et al., *The use of time-variant EEG Granger causality for inspecting directed interdependencies of neural assemblies*, in *J Neurosci Methods*2003. p. 27-44.
25. Marinazzo, D., et al., *Nonlinear connectivity by Granger causality*, in *Neuroimage*2011. p. 330-8.
26. Seth, A.K., *A MATLAB toolbox for Granger causal connectivity analysis*, in *J Neurosci Methods*2010. p. 262-73.
27. Barrett, A.B., et al., *Granger causality analysis of steady-state electroencephalographic signals during propofol-induced anaesthesia*, in *PLoS ONE*2012. p. e29072.

28. Cohen, A.L., et al., *Defining functional areas in individual human brains using resting functional connectivity MRI*, in *Neuroimage*2008. p. 45-57.
29. Cui, X., et al., *A quantitative comparison of NIRS and fMRI across multiple cognitive tasks*, in *Neuroimage*2010.
30. Nummenmaa, L., et al., *Connectivity analysis reveals a cortical network for eye gaze perception*, in *Cereb Cortex*2010. p. 1780-7.
31. Welch, P.D., *The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms*. Audio and Electroacoustics, IEEE Transactions on. **15**(2).
32. Wiener, N., *The theory of prediction*. In *Modern Mathematics for Engineers*. New York: McGraw-Hill, 1956. **1**.
33. Granger, C.W.J., *Investigating causal relations by econometric models and cross-spectral methods*. *Econometrica* 1969. **37**: p. 424-438.
34. Hillebrand, A., et al., *Frequency-dependent functional connectivity within resting-state networks: an atlas-based MEG beamformer solution*, in *Neuroimage*2012. p. 3909-21.
35. Hipp, J.F., et al., *Large-scale cortical correlation structure of spontaneous oscillatory activity*, in *Nat Neurosci*2012. p. 884-90.
36. Lachaux, J.P., et al., *Measuring phase synchrony in brain signals*, in *Human brain mapping*1999. p. 194-208.
37. Barrie W. Jervis, M.J.N., Terence E. Johnson, Elaine Allen, And Nigel R. Hudson, *A fundamental investigation of the composition of auditory evoked potentials*. IEEE Transactions on Biomedical Engineering, 1983. **BME-30**(1).
38. Tallon-Baudry, C., et al., *Oscillatory gamma-band (30-70 Hz) activity induced by a visual search task in humans*, in *J Neurosci*1997. p. 722-34.
39. Grossman, A.e.a., *Reading and understanding continuous wavelets transforms In: Wavelets, time-frequency methods and phase space*. Berlin: Springer, 1989.
40. Fisher, N.I., *Statistical Analysis of Circular Data*. Press Syndicate of the University of Cambridge. New York, NY, 1993.
41. Johnson, N.L., Kotz, Samuel, & Kemp, Adrienne W., *Univariate Discrete Distributions, Second Edition*. 1992: Wiley.
42. Williams, R.W. and K. Herrup, *The control of neuron number*. *Annu Rev Neurosci*, 1988. **11**: p. 423-53.

43. Varsou, O., M.J. Macleod, and C. Schwarzbauer, *Functional connectivity magnetic resonance imaging in stroke: an evidence-based clinical review*. Int J Stroke, 2013.
44. Yu, R., et al., *Enhanced Functional Connectivity between Putamen and Supplementary Motor Area in Parkinson's Disease Patients*. PLoS One, 2013. **8**(3): p. e59717.
45. Tijms, B.M., et al., *Alzheimer's disease: connecting findings from graph theoretical studies of brain networks*. Neurobiol Aging, 2013.
46. Taniwaki, T., et al., *Disrupted connectivity of motor loops in Parkinson's disease during self-initiated but not externally-triggered movements*. Brain Res, 2013.
47. Blain-Moraes, S., et al., *Altered Cortical Communication in Amyotrophic Lateral Sclerosis*. Neurosci Lett, 2013.
48. Biggs, N.L., E.; Wilson, R, *Graph Theory*. Oxford University Press, 1986.
49. Bondy, J.A.M., U.S.R., *Graph Theory*. Springer. 2008.
50. K., B., *Vergleichende Lokalisationslehre der Grosshirnrinde*. 1909.
51. Maris, E. and R. Oostenveld, *Nonparametric statistical testing of EEG- and MEG-data*, in *J Neurosci Methods* 2007. p. 177-90.
52. B. Efron, R.J.T., *An Introduction to the Bootstrap*. 1994: Chapman and Hall/CRC.
53. Dwass, M., *Modified Randomization Tests for Nonparametric Hypotheses*. The Annals of Mathematical Statistics, 1957. **28**: p. 181-187.
54. Bickel, P.M.a.V.Z., W.R., *Asymptotic expansion for the power of distribution-free tests in the two-sample problem*. Annals Of Statistics, 1987. **6**: p. 987-1004.
55. Gandy, A., *Sequential implementation of Monte Carlo tests with uniformly bounded resampling risk*. Journal of the American Statistical Association 2009. **104**(488): p. 1504–1511.
56. Hamalainen, M.S. and R.J. Ilmoniemi, *Interpreting magnetic fields of the brain: minimum norm estimates*. Med Biol Eng Comput, 1994. **32**(1): p. 35-42.
57. Hamalainen, M.S., *Functional localization based on measurements with a whole-head magnetometer system*. Brain Topogr, 1995. **7**(4): p. 283-9.
58. Hamalainen, M.S., *Magnetoencephalography: a tool for functional brain imaging*. Brain Topogr, 1992. **5**(2): p. 95-102.

59. L. John Greenfield, J.D.G., Paul R. Carney, *Reading EEGs: A Practical Approach*. 2009: Lippincott Williams & Wilkins.
60. Okada, Y., *Neurogenesis of evoked magnetic fields*. New York: Plenum Press, 1983: p. 399-408.
61. Barth, D.S., W. Sutherling, and J. Beatty, *Intracellular currents of interictal penicillin spikes: evidence from neuromagnetic mapping*. Brain Res, 1986. **368**(1): p. 36-48.
62. Ghuman, A.S., et al., *The effects of priming on frontal-temporal communication*, in *Proc Natl Acad Sci USA*2008. p. 8405-9.
63. Gow, D.W., et al., *Lexical influences on speech perception: a Granger causality analysis of MEG and EEG source estimates*, in *Neuroimage*2008. p. 614-23.
64. Lin, F.-H., et al., *Spectral spatiotemporal imaging of cortical oscillations and interactions in the human brain*, in *Neuroimage*2004. p. 582-95.
65. *MNE-manual-2.6*, 2009. p. 1-344.
66. Chiong, W., et al., *The salience network causally influences default mode network activity during moral reasoning*. Brain, 2013.
67. Ding, W.N., et al., *Altered default network resting-state functional connectivity in adolescents with internet gaming addiction*. PLoS One, 2013. **8**(3): p. e59902.
68. Amianto, F., et al., *Intrinsic Connectivity Networks Within Cerebellum and Beyond in Eating Disorders*. Cerebellum, 2013.
69. Guldenmund, P., et al., *Thalamus, brainstem and salience network connectivity changes during mild propofol sedation and unconsciousness*. Brain Connect, 2013.
70. Wang, J.X., et al., *Changes in Task-Related Functional Connectivity across Multiple Spatial Scales Are Related to Reading Performance*. PLoS One, 2013. **8**(3): p. e59204.
71. Biswal, B.B., *Resting state fMRI: a personal history*. Neuroimage, 2012. **62**(2): p. 938-44.
72. Fox, M.D. and M.E. Raichle, *Spontaneous fluctuations in brain activity observed with functional magnetic resonance imaging*, in *Nat Rev Neurosci*2007. p. 700-11.
73. Boly, M., et al., *Intrinsic brain activity in altered states of consciousness: how conscious is the default mode of brain function?* Ann N Y Acad Sci, 2008. **1129**: p. 119-29.
74. Toppi, J., et al., *Describing relevant indices from the resting state electrophysiological networks*. Conf Proc IEEE Eng Med Biol Soc, 2012. **2012**: p. 2547-50.

75. Wang, L., et al., *Cortical networks of hemianopia stroke patients: a graph theoretical analysis of EEG signals at resting state*. Conf Proc IEEE Eng Med Biol Soc, 2012. **2012**: p. 49-52.
76. Kay, B.P., et al., *Reduced default mode network connectivity in treatment-resistant idiopathic generalized epilepsy*. Epilepsia, 2013. **54**(3): p. 461-70.
77. Madhavan, D., E. Heinrichs-Graham, and T.W. Wilson, *Whole-brain functional connectivity increases with extended duration of focal epileptiform activity*. Neurosci Lett, 2013.
78. Ghanbari, Y., et al., *Dominant component analysis of electrophysiological connectivity networks*. Med Image Comput Comput Assist Interv, 2012. **15**(Pt 3): p. 231-8.
79. Strafella, A.P., *Anatomical and functional connectivity as a tool to study brain networks in Parkinson's disease*. Mov Disord, 2013. **28**(4): p. 411-2.
80. Hacker, C.D., et al., *Resting state functional connectivity of the striatum in Parkinson's disease*. Brain, 2012. **135**(Pt 12): p. 3699-711.
81. Tessitore, A., et al., *Default-mode network connectivity in cognitively unimpaired patients with Parkinson disease*. Neurology, 2012. **79**(23): p. 2226-32.
82. Canuet, L., et al., *Resting-state network disruption and APOE genotype in Alzheimer's disease: a lagged functional connectivity study*. PLoS One, 2012. **7**(9): p. e46289.
83. de Haan, W., et al., *Activity dependent degeneration explains hub vulnerability in Alzheimer's disease*. PLoS Comput Biol, 2012. **8**(8): p. e1002582.
84. Li, R., et al., *Alterations of directional connectivity among resting-state networks in Alzheimer disease*. AJNR Am J Neuroradiol, 2013. **34**(2): p. 340-5.
85. Shepherd, G.M., *Corticostriatal connectivity and its role in disease*. Nat Rev Neurosci, 2013. **14**(4): p. 278-91.
86. Stevenson, R.A., *Using functional connectivity analyses to investigate the bases of autism spectrum disorders and other clinical populations*. J Neurosci, 2012. **32**(50): p. 17933-4.
87. Yu, R., et al., *Frequency dependent alterations in regional homogeneity of baseline brain activity in schizophrenia*. PLoS One, 2013. **8**(3): p. e57516.
88. Shim, W.H., et al., *Frequency distribution of causal connectivity in rat sensorimotor network: resting-state fMRI analyses*. J Neurophysiol, 2013. **109**(1): p. 238-48.

89. Ghuman, A.S., R.N. van den Honert, and A. Martin, *Interregional neural synchrony has similar dynamics during spontaneous and stimulus-driven states*. *Sci Rep*, 2013. **3**: p. 1481.
90. S. Taulu, M.K., and J. Simola, *Suppression of interference and artifacts by the signal space separation method*. *Brain Topography*, 2004 **16**(4): p. 269-275.
91. S. Taulu, a.M.K., *Presentation of electromagnetic multichannel data: the signal space separation method*. *Journal of Applied Physics*, 2005. **97**(12).
92. S. Taulu, J.S., and M. Kajola, *Applications of the signal space separation method*. *IEEE Transactions on Signal Processing*, 2005. **53**(9): p. 3359-3372.
93. S. Taulu, a.J.S., *Spatiotemporal signal space separation method for rejecting nearby interference in MEG measurements*. *Physics in Medicine and Biology*, 2006. **51**: p. 1759-1768.
94. Greicius, M.D., et al., *Resting-state functional connectivity reflects structural connectivity in the default mode network*, in *Cereb Cortex*2009. p. 72-8.
95. Smyser, C.D., A.Z. Snyder, and J.J. Neil, *Functional connectivity MRI in infants: exploration of the functional organization of the developing brain*. *NeuroImage*, 2011. **56**(3): p. 1437-52.
96. Carter, A.R., G.L. Shulman, and M. Corbetta, *Why use a connectivity-based approach to study stroke and recovery of function?* *NeuroImage*, 2012. **62**(4): p. 2271-80.
97. Carbon, M. and R.M. Marié, *Functional imaging of cognition in Parkinson's disease*. *Current opinion in neurology*, 2003. **16**(4): p. 475-80.
98. Wurina, Y.F. Zang, and S.G. Zhao, *Resting-state fMRI studies in epilepsy*. *Neuroscience bulletin*, 2012. **28**(4): p. 449-55.
99. Li, T.Q. and L.O. Wahlund, *The search for neuroimaging biomarkers of Alzheimer's disease with advanced MRI techniques*. *Acta radiologica* (Stockholm, Sweden : 1987), 2011. **52**(2): p. 211-22.
100. Minshew, N.J. and T.A. Keller, *The nature of brain dysfunction in autism: functional brain imaging studies*. *Current opinion in neurology*, 2010. **23**(2): p. 124-30.
101. Van Essen, D.C., et al., *The Human Connectome Project: a data acquisition perspective*. *NeuroImage*, 2012. **62**(4): p. 2222-31.
102. Greicius, M., *Resting-state functional connectivity in neuropsychiatric disorders*. *Current opinion in neurology*, 2008. **21**(4): p. 424-30.

103. Sperling, R.A., et al., *Functional alterations in memory networks in early Alzheimer's disease*. Neuromolecular medicine, 2010. **12**(1): p. 27-43.
104. Fornito, A. and E.T. Bullmore, *What can spontaneous fluctuations of the blood oxygenation-level-dependent signal tell us about psychiatric disorders? Current opinion in psychiatry*, 2010. **23**(3): p. 239-49.
105. Smith, S.M., et al., *Correspondence of the brain's functional architecture during activation and rest*. Proceedings of the National Academy of Sciences of the United States of America, 2009. **106**(31): p. 13040-5.
106. He, B.J., et al., *The temporal structures and functional significance of scale-free brain activity*, in *Neuron* 2010. p. 353-69.
107. de Pasquale, F., et al., *Temporal dynamics of spontaneous MEG activity in brain networks*, in *Proc Natl Acad Sci USA* 2010. p. 6040-5.
108. Arieli, A., et al., *Dynamics of ongoing activity: explanation of the large variability in evoked cortical responses*. Science (New York, NY), 1996. **273**(5283): p. 1868-71.
109. Damoiseaux, J.S. and M.D. Greicius, *Greater than the sum of its parts: a review of studies combining structural connectivity and resting-state functional connectivity*. Brain structure & function, 2009. **213**(6): p. 525-33.
110. Sporns, O., G. Tononi, and R. Kötter, *The human connectome: A structural description of the human brain*, in *PLoS Comput Biol* 2005. p. e42.
111. Roebroeck, A., E. Formisano, and R. Goebel, *The identification of interacting networks in the brain using fMRI: Model selection, causality and deconvolution*, in *Neuroimage* 2011. p. 296-302.
112. Gregoriou, G.G., et al., *High-frequency, long-range coupling between prefrontal and visual cortex during attention*, in *Science* 2009. p. 1207-10.
113. de Haan, W., et al., *Disrupted modular brain dynamics reflect cognitive dysfunction in Alzheimer's disease*. NeuroImage, 2012. **59**(4): p. 3085-93.
114. Westlake, K.P., et al., *Resting state  $\alpha$ -band functional connectivity and recovery after stroke*. Experimental neurology, 2012. **237**(1): p. 160-9.
115. Strogatz, S.H., *Exploring complex networks*. Nature, 2001. **410**(6825): p. 268-76.
116. Stam, C.J., et al., *Graph theoretical analysis of magnetoencephalographic functional connectivity in Alzheimer's disease*. Brain : a journal of neurology, 2009. **132**(Pt 1): p. 213-24.

117. Lu, H. and E.A. Stein, *Resting state functional connectivity: Its physiological basis and application in neuropharmacology*. Neuropharmacology, 2013.
118. van den Heuvel, M.P., et al., *Aberrant frontal and temporal complex network structure in schizophrenia: a graph theoretical analysis*, in *J Neurosci* 2010. p. 15915-26.
119. Siebenhühner, F., et al., *Intra- and inter-frequency brain network structure in health and schizophrenia*. PLoS ONE, 2013. **8**(8): p. e72351.
120. Werner, C.J., et al., *Altered resting-state connectivity in Huntington's Disease*. Human brain mapping, 2013.
121. Bavelas, A., *A mathematical model for group structure*. Anthropology, 1948. **7**: p. 16-39.
122. Binnewijzend, M.A., et al., *Brain network alterations in Alzheimer's disease measured by Eigenvector centrality in fMRI are related to cognition and CSF biomarkers*. Human brain mapping, 2013.
123. Wink, A.M., et al., *Fast eigenvector centrality mapping of voxel-wise connectivity in functional magnetic resonance imaging: implementation, validation, and interpretation*. Brain connectivity, 2012. **2**(5): p. 265-74.
124. Lohmann, G., et al., *Eigenvector centrality mapping for analyzing connectivity patterns in fMRI data of the human brain*. PLoS ONE, 2010. **5**(4): p. e10232.
125. Leicht, E.A. and M.E. Newman, *Community structure in directed networks*. Physical review letters, 2008. **100**(11): p. 118703.
126. Milo, R., et al., *Network motifs: simple building blocks of complex networks*. Science (New York, NY), 2002. **298**(5594): p. 824-7.
127. Wang, J., et al., *Parcellation-dependent small-world brain functional networks: a resting-state fMRI study*. Human brain mapping, 2009. **30**(5): p. 1511-23.
128. Dale, A.M., B. Fischl, and M.I. Sereno, *Cortical surface-based analysis. I. Segmentation and surface reconstruction*. NeuroImage, 1999. **9**(2): p. 179-94.
129. Fischl, B., M.I. Sereno, and A.M. Dale, *Cortical surface-based analysis. II: Inflation, flattening, and a surface-based coordinate system*. NeuroImage, 1999. **9**(2): p. 195-207.
130. MacQueen, J., *Some methods for classification and analysis of multivariate observations*. 1967, Proc. 5th Berkeley Symp. Math. Stat. Probab., Univ. Calif. 1965/66, 1, 281-297 (1967).
131. Caliński, T., *Dendrogram*. Encyclopedia of Biostatistics, 2005.

132. Hämäläinen, M., Hari, R., Ilmoniemi, R.J., Knuutila, J., Lounasmaa, O.V., *Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the human brain*. Rev. Mod. Phys., 1993. **65**: p. 413–497.
133. Gramfort, A., et al., *MNE software for processing MEG and EEG data*. Neuroimage, 2014. **86**: p. 446-60.
134. Lin, F.H., et al., *Assessing and improving the spatial accuracy in MEG source localization by depth-weighted minimum-norm estimates*. NeuroImage, 2006. **31**(1): p. 160-71.
135. Catani, M., D.K. Jones, and D.H. ffytche, *Perisylvian language networks of the human brain*. Annals of neurology, 2005. **57**(1): p. 8-16.
136. Daunizeau, J. and K.J. Friston, *A mesostate-space model for EEG and MEG*. Neuroimage, 2007. **38**(1): p. 67-81.
137. Olier, I., N.J. Trujillo-Barreto, and W. El-Deredy, *A switching multi-scale dynamical network model of EEG/MEG*. Neuroimage, 2013. **83**: p. 262-87.