MÁSTERES de la UAM

Facultad de Psicología / 16-17

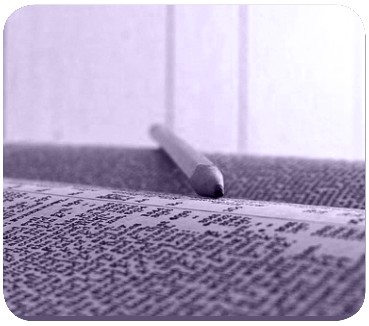
Metodología de las Ciencias del Comportamiento y de la Salud

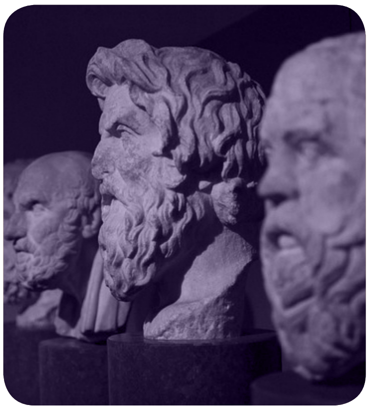
UAM
UNIVERSIDAD AUTONOMA
DE MADRID

Campus Internacional
excelencia UAM+CSIC

Brain network's synchronization measures: a statistical approach

Ignacio Echegoyen Blanco





UAM
EDICIONES

UNIVERSIDAD COMPLUTENSE
MADRID

UNIVERSIDAD AUTÓNOMA
DE MADRID

Master's Degree in Methodology of Behavioral and Health Sciences

BRAIN NETWORK'S SYNCHRONIZATION MEASURES: A STATISTICAL APPROACH

Ignacio Echegoyen

supervised by
Dr. Javier Revuelta

September 7, 2017

Contents

Abstract

Synchronization Measures (SM) are a common tool to quantify the relationship between brain regions, and serve as a first step to build functional networks, as studied in the field of Complex Networks. SM offer robust and reliable results only when the signal exhibit certain mathematical properties: linearity, stationarity and a good Signal to Noise Ratio. Due to filtering procedures, brain signals are sampled, and it is common practice to rejoin them to obtain a larger dataset. Here we present four SM, measured over the timeseries as a whole, and broken into samples, checking their mathematical properties, and looking for differences in SM depending on whether or not those properties are met. It is concluded that is preferable to conduct connectivity analysis over samples than over the whole time series, although in any case the signal meets completely the required properties.

1 | Introduction: Brain Networks

In the present work we will show general concepts from the field of brain networks and brain data analysis, and a statistical analysis on synchronization measures will be conducted. Synchronization measures are used to build functional brain networks from neural data, as we will show, and are the main topic of this work.

Functional brain networks are studied from the complex networks perspective, which is based on statistical physics and graph theory. To get insights in the topic and conduct further analysis, first, we will define different brain networks (or graphs), their uses, origins (from different types of neural data) and representation. Then, a brief introduction to the foundations of complex sciences and nonlinearity will be provided. We will also present four synchronization measures commonly used to build functional brain networks, and, at last, statistical comparisons between them will be conducted.

It is common practice to conduct functional network analysis, given as a matter of fact that the basic conditions needed to get robustness and reliability are met. The goal of this study is to assess those mathematical assumptions under which coordination measures are based, to better understand the relationships between the required conditions and coordination measures.

1.1 | Dynamic, Non-Linear, Complex Systems

Classically, science has been articulated through Reductionist Materialism, the philosophical position that states that everything is material, and reductible to its components. This perspective is the landmark of the western cosmology, the way in which the universe and its living forms are studied and analyzed. The components that comprise a system (and the system itself) are mechanisms, thus knowing the parts and their rules of motion is enough to understand a system, predict it, and describe its previous states, because the system is nothing more than the sum of its parts. This conceptual framework - a clockwork universe - has been applied to every system studied in the known universe, from rocks to human beings. It was set by Kepler, Galileo and Newton, and develops in parallel to Descartes' (and before him, Plato's) onthology. It is the signature of Rationalism. Its scientific counterpart is characterized by (linear) causality, determinism, continuity and reversibility (Érdi, 2008). Throughout history, it seemed that all kind of dynamic phenomena could be reduced to mechanical motions, that Newtonian principles could unify every motion, from planet's orbits to human behavior; that is, that the same Laws of Nature govern every phenomena.

Linear causality, determinism, continuity and reversibility are appropriate concepts to describe simple systems, where the linear approximation is good enough, and are milestones in classical mechanics. But nature doesn't act this way. Whenever parts of a system interfere, cooperate o compete, there are nonlinear interactions going on. Fluctuations from the average cannot be neglected, and the principle of superposition, by which a linear system can be solved breaking it into solveable parts to later recombine them, fails sistematically. The linear aproximation is very bad, and new mathematical tools are needed to characterize the system (Strogatz, 1994). There are uncountable examples of nonlinear systems, where a holistic approach can offer better results. It is not only a matter of nonlinearity, as, for example, linear regressions can modelize power-two equations, linearizing the system with the link function. Neither is only a matter of the number of variables, as many problems with infinite variables (or degrees of freedom) are treatable with linear mathematical tools [Fourier decomposition series, for instance; (Strogatz, 1994)]. The gist is the emergence of complexity, of new properties not accounted by the individual parts composing the system. That is precisely what the science of complexity studies. Quoting Érdi: "the science of complexity suggests that while life is in accordance with the laws of physics, physics cannot predict life. Therefore, in addition to reductionism, a more complete understanding of complex dynamical systems requires some holism" (Érdi, 2008, *p.*4). Hence, it is said that the whole is greater than the sum of its parts, and the studied systems are called to be complex. Here, complexity is put as an opposite to to simplicity. An airplane, though complicated, is not complex, because its parts don't interact, and there are not emergent properties (Martínez Huartos, 2015). But, how do system's properties arise from interactions among the system's components? This is a never-ending quest in Complex Sciences. In this context, last century's sciences developed many new concepts - or renewed older ones - , in an attempt to explain this macro-behaviors, inexplicable from a micro perspective. Is in this direction that complexity is defined.

Complexity is a key concept in the study of dynamical, non-linear systems, from physical oscillators to brain function, economics (econophysics), social emerging patterns, disease spreading or opinion formation, to name a few, because it is known that when systems are complex, new properties emerge. Mathematically, complexity has to do with the ratio of order-randomness, and has been studied, mainly, from two perspectives. The first one, set by Kolmogorov, accounts for the (im)possibility to reduce any given string of numbers to a pattern. If the reduction is maximum or minimum, the pattern is completely ordered or completely random, thus not complex. The second approach explores this idea of randomness-order from a structural perspective, in graphs. Both extremes (complete randomness and complete order) are considered as not complex, and are relatively easy to produce artificially. As expected, the systems complex sciences deal with are in-between, not purely random nor ordered. Structural complexity is measured statistically, quantifying the difference between a system's characteristic's distribution to the expected if it were completely random, and comparing it to it's entropy (that has to do with the degrees of freedom and disorder of the system). This concepts have been used as a guide to build models of systems and networks, that serve as a starting point for many real systems simulations. The most commonly known examples of this are the Barabási-Albert Network, and the Erdos-Renyi Network, both departing from lattice or completely random networks and evolving to plausible real-world networks. (Sporns, 2011; Érdi, 2008).

The structure of a system has to do with the way in which its elements are arranged and connected. The dynamics of a system is the precise way in which those elements interact and exchange information, leading to the emergence of new functions and macro states impossible to explain studying only the components of the system independently. The big mystery is how simple elements, interacting together, can adapt to a changing environment, self-organizing and self-correcting to seek stable goals on a macro level.

As this perspective evolved, structural properties, shared by networked systems and not dependant on the particularities of the system, were unveiled. One of them was discovered by social psychologist Stanley Milgram, and states that we live in a small world. It means that the friends of the friends of the friends of my friends turns out to be the whole population (Érdi, 2008). This is due to the formation of clusters or communities (as the friends of my friends tend to be also my friends) and low shortest paths (on average), which is the number of steps needed to go from one node to any other (Newman, 2010). Another ubiquitous network property is the extremely high variance of node degree (number of nodes a node is connected to). Some nodes has very few connections and are almost isolated. A few are extremely highly connected, the so-called *hubs*. The vast majority of nodes are somewhere in between. These networks exhibit power-law degree distribution, and are often called scale-free, as they have no characteristic scale for node degree: every kind of node is present, and there is no such thing as a “typical node” (Érdi, 2008). At first, these and other common properties were thought to be present only in some well known systems. But, later on, many other non-related networks (the brain, for instance) were discovered to be also organized in such a manner, with nodes forming communities, some of them extremely connected (hubs), and every node being reached by any other node in only a few steps. The brain is not an exception: from the complexity of the neuron level to that of the graph structure at whole-brain level, brain networks reveal high small-worldness, power-law degree distributions and hubs (even hubs of hubs, the “rich club”), related to association and integration processes (Papo et al., 2015).

1.2 | Networks

Complex Sciences, concretely complex networks analysis, are based on graph theory and statistical mechanics, and study systems composed by interacting parts. Examples of this systems can be social interactions, bird flock formation, or the brain. In such cases, systems’ functions, although conditioned by the structure of the system, emerge from it’s components’ interactions, and no unique agent can be identified as direct cause of the observed functions and behaviors. That is the reason why Complex Sciences focuses on whole systems, and not on the individual parts that compose it. In order to do that, systems’ components and relations are projected on graphs.

A graph is an abstraction of any N given elements, represented by points (nodes), connected by L lines (edges). In its simplest form, it is a collection of points joined together in pairs by lines (Newman, 2010; Estrada, 2011)(example in *figure 1.1*). Graphs can be used every time a system composed of individual parts or components linked together in some way is represented or analyzed. Thus, it is useful to visualize and analyze a broad range of physical, social or biological systems. Depending on the system, a graph may represent interactions among individuals, where each person is a node, and interactions between them are edges (this can be applied to physical or virtual social networks), public transport usage, where each station is a node, and the edges are the connections between them, or any other relation among any pair of elements. Graphs can be of very different nature (that of the system represented), and notation also changes depending on the field of study. Nodes and edges are called *sites* and *bonds* in physics, *nodes* and *links* in computer science, or *actors* and *ties* in sociology (Newman, 2010), to name a few.

The concept of graph originally comes from Euler, who tried to solve the popular problem of the seven bridges of Königsberg. The town was set on both sides of the Pregel River, with two large islands connected to each other by seven bridges. The problem was to devise a walk through both mainland portions, crossing the seven bridges once, and only once. Euler proved it was impossible, and to do so, he represented the portions of land as nodes or vertices, each one connected to one another by edges or lines representing the bridges. This was the foundation of graph theory (Bullmore & Sporns, 2009; Sporns, 2011).

Some fields of study are more focused on the elements; others, on the link between them. In any case, there is a third aspect to these interacting systems, whose importance, although neglected sometimes, is almost always crucial to the behavior of the system: the *pattern* of connections between components (Newman, 2010). It is known that the pattern of connections has deep effects on the overall functioning of the network, contributing to the emergence of global characteristics observed in the graph (Érdi, 2008; Strogatz, 2001). There are complex relations between structure and function. A particular structure favour the emergence of concrete patterns, and not others, and those patterns are closely related to the emergence of global functions the system may exhibit. The field of complex networks studies, among other things, these relations between structure, the emergence of patterns, and function, with special attention to the importance of characterizing the anatomy of the network (as structure always affects function).

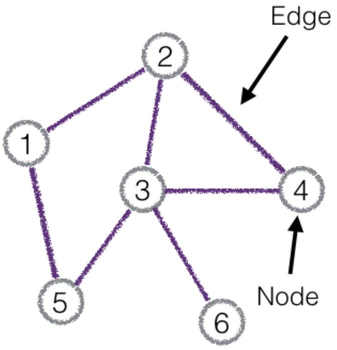


Figure 1.1: Simple graph.

Complex Networks analysis, based on this graph perspective, and making use of statistics, tries to explain global properties and functions emerging from individual interactions between the elements of the system (Érdi, 2008). From microscopic phenomena, hierarchically organised, emerge statistical, macroscopic properties, impossible to predict without a wholistic framework analysis (Papo et al., 2015). Nevertheless, it is always a simplified version of reality, an aproximation, thus some information is always lost (Newman, 2010).

Nowadays, graph/network analysis are common practice in biology, sociology, chemistry, physics and neuroscience. A molecule, a diagram of interactions between people, protein interactions and many other phenomena can be represented by graphs. Despite this conceptualization has led to great advances in every field, many systems are still unpredictable. For instance, *Caenorhabditis Elegans*' connectome (map of neural wiring) is completely identified, yet its behavior remains impossible to predict (Sporns, 2011).

1.2.1 Classification and notation

Networks can be classified attending to several characteristics of nodes or edges. After nodes are identified (depending on the scale of observation), and edges calculated, the network can be represented in a matrix, whose rows and columns correspond to nodes, and the cells inside contain the value of the edge that links those nodes. In the present work we will show only a few variations on edge characteristics, and we'll provide the most general and common classification of networks and its notation, focusing on its applications on the study of the brain.

Edges represent a relationship between a pair of nodes, and can be **binary** or **unweighted** if have a value of zero or one, or **weighted** if the value has a range of decimal intervals (Newman, 2010). Depending on the number of edges equal to zero, the matrix (graph) is said to be **dense** or **sparse**. Edges can also be **directed** or **undirected** depending on whether direction (which implies causality) from one node to another is imposed. When that happens, it is assumed that information flows from one node to another. That is, one node's output is another node's input. It is also possible that a node's output serves as an input for itself, generating a loop. The presence or not of self-edges determines if the network is **cyclic** or **acyclic**.

To represent a network in a adjacency matrix, nodes are numbered and assigned to a column and a row, and edges, representing the value of the connection between nodes, are placed in the cell of the matrix through which that node's row comes across the others. We can define any *adjacency matrix* A_{ij} with elements a_{ij} as follows:

$$a_{ij} = \begin{cases} w & \text{if there is an edge between vertices } i \text{ and } j, \\ \\ 0 & \text{otherwise.} \end{cases} \tag{1.1}$$

In the simplest case, when the graph is unweighted (or binary) and undirected, w can only have a value of 1 (if there is an edge between nodes) or 0 (if there is not). For example, the adjacency matrix of the graph represented in *figure 1.1* is:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{1.2}$$

Where the element in column's ith position corresponds to the ith node, the element in row's jth position corresponds to the jth node, and the ijth position in the matrix correspond to the edge between them.

It is always a square matrix, and in undirected graphs, the matrix is also symmetric, since if there is an edge between i and j then there is an edge between j and i. Symmetry is a very desirable property when calculating further network properties, and it reduces the time needed to compute the matrix by half. It is also worth noting that if the network has no self-edges, all elements in the diagonal are zero, and it is called an acyclic network (Newman, 2010).

Nevertheless, things can be much more complicated. Graphs can be weighted, directed, and with self edges; its adjacency matrix will not be symmetric, and diagonal elements could have values different from zero. In such case, w (in *def. 1.1*) is the weight of the edge from i to j, and will have a value different from 1. Instead, w will have a range that depends on the concrete calculation of the correlation between nodes. In this case, the matrix is commonly named *weighted connectivity matrix.* Also, if we allow $j = i$ then diagonal elements are non-zero and we have a cyclic network. The adjacency/connectivity matrix, then, captures the direction and the weight of the interaction or coordination between nodes.

1.3 | Networks of the Brain

Analyzing the brain in a complex-network fashion requires the projection of anatomical or dynamical brain properties in a graph. Nodes can be neurons, groups of neurons, complete brain regions, or sensors (that measure brain's activity), depending on the scale of observation. In our case, nodes will be sensors from a MEG machine, and links will be the value from coordination measures, that capture dynamical synchronization between nodes. The rationale is that if two brain regions are working together, there must be similarities in their registered activity. Due to the fact that neural coupling processes are not well understood, and probably work under several mechanisms, similarities can be found in the frequency domain, in amplitude, and/or in phase and different measures capture different aspects or mechanisms of synchronization. After recording the brain activity through sensors, coordination measures are used as a mean to quantify the relationship between nodes, and will be the weight of the link between those nodes. Depending on whether the network is capturing the anatomical structure of the brain, or its functional activity, we can define *Anatomical Connectivity* (AC) and *Functional Connectivity* (FC). When the edge from one node to another is not the same in reverse the network is said to be *directed* (Estrada, 2011; Newman, 2004). This implies causality inference, and the interpretation is about information flowing from one region to another, thus one region's state causing others' state, at least partially. That is called *Effective Connetivity* (EC).

Common measures of FC are *Phase Lag Index*, *Coherence* or *Mutual Information* (Bullmore & Sporns, 2009), and are based on properties of the signal, being those physical (amplitude, phase), or statistical (entropies). FC studies correlations, the similarities of phenomena occurring at the same time, or the extent to which phenomena co-occur. Subsequently, the networks built in FC are "virtual"; that is, the network does not exist physically, as it is based on dynamic processes in the brain, on the association between regions doing similar things at the same time (synchronously). An adjacency matrix of a Functional Network is nothing more than a snapshot of a brain dynamic process that changes over time. The relationship between Anatomical Networks, that are the underlying physical substrate of dynamical processes, and Functional Networks, extracted from the association of those processes, is a complete field in itself (Sporns, 2011).

Common measures of EC are *Granger Causality* (parametric) or *Transfer Entropy* [non-parametric (Maestú, Pereda, & del Pozo, 2015)]. This statistic measures try to quantify the amount of directed (time-asymmetric) transfer of information between two random processes [time series obtained by sensors, the brain data signals to be analyzed; Friston, Moran, and Seth (2013), Seth, Barrett, and Barnett, 2015]. It is important to note that in this particular case (causal modelling for neural data series), EC is based on Norbert Wiener's concept of causality, that states the importance of temporal succession when stating causality. He defined causality in a statistical manner: let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be two simultaneous measured time-dependent magnitudes. If the accuracy with which we predict X's future values increases more when adding past information from Y, in comparison to predictions provided only by past values of X, then we can say there is a causal relationship from Y to X (Wiener, 1956, cited in Maestú et al., 2015). That is, to some exent, Y causes X. This concept is important because is based on statistical logic, and avoids obscure assumptions on physical causality in the brain dynamics, as measured by MEG/EEG, for instance. Following this line of thought, Granger formalised the concept in the context of linear regression modelling (L. Barnett & Seth, 2014; L. Barnett & Seth, 2011). The idea behind Granger Causality and Transfer Entropy is fundamentally the same: a linear regression model is adjusted estimating the coefficients with past direct values of Y to predict X (Granger Causality, parametric), or with past entropies of Y, to predict X (Transfer Entropy, not parametric). A general revision of this terms and mathematical tools can be found in Maestú et al. (2015), and a deeper one in Cohen (2014) or Friston et al. (2013).

Although directed networks are probably more accurate to brain's reality, it is more common to work with undirected ones (Sporns, 2011). It is more plausible that brain regions connect to each other in different ways, and that information flows differentially in reverse. But, when networks are undirected, the adjacency matrix is symmetric, which is very convenient for connectivity calculations and makes computation notably faster. Undirected networks are reflecting synchronization processes in the brain, regions working together. Directed networks are reflecting information flow, and probably dominance, from one region to another.

Edges will take a value in the range of the coordination measure used to capture the relationship among nodes. Commonly, normalizations and transformations are carried out to set that range between 0 and 1, but other regular ranges are -1 to 1, or 0 to ∞. The interpretation of the results vary depending on the range of the measure and the correspondance between the synchronized/associated characteristic and the brain process behind it. This makes interpretation difficult, and it is not clear what a value of, for example, 18.9 means when edges can take values up to ∞ and nodes are sensors and not brain regions directly. When edges take the value of the synchronization measure, the graph is weighted. In many contexts, it is common practice to binarize weights for calculation and interpretation purposes (Sporns, 2011), leading to unweighted or binary networks.

1.4 | Neural Data: MEG Registers

In the present work, neural data obtained from Magnetoencephalography (MEG) will be analyzed through several types of coordination measures, whose results will form a connectivity matrix from which to get network parameters.

MEG is a well established technique based upon brain's magnetic fields. It is not exempt from problems of noise, volume conduction, signal interferences and many other technical issues. Given the prize of the equipment, its use is far less common than EEG, although it presents many advantages over the latter. In this section we will discuss brain magnetic fields origins, as measured by MEG, properties of the machine, and some pitfalls needed to be taken into account while recording magnetic data. We will also review - roughly - the problem of sources reconstruction, an open question in brain data signals.

In brain data techniques (fMRI, (M/E)EG, DTI, etc) there is always a trade-off between spatial and temporal resolution (Carretie, 2011). Haemodynamic-based machines, such as fMRI, resolve spatial resolution up to 3mm (and increasing with technical developments), but, given the slow speed of the underlying mechanisms measured (metabolic and water support to working brain areas), its temporal resolution cannot go further 1 second. On the other hand, techniques based on magnetic fields or electric currents, as MEG and EEG respectively, can resolve extremely accurate temporal intervals, at the cost of spatial resolution (Maestú et al., 2015). For example, at a sampling rate of 1000 Hz, a thousand values in fT (10^{-15} tesla) are measured (in the case of magnetic fields, with MEG), in one second. The records are obtained noninvasively, and in MEG/EEG signals, brain activity can be broken down to the milisecond, providing an almost real-time snapshot. Given this framework, some techniques are more appropiate to assess certain mental processes than others. If the studied processes are inherently dynamical (e.g., attentional shifts among probe stimuli), MEG and EEG are the preferred choice. On the contrary, if the process has to do with structural connections or slow mental tasks (e.g., sustained attention), fMRI or DTI are more appropiate. There is nothing as the perfect solution, and due to each technique virtues, the best option depends on the task. Regarding the brain, it is known that it works at several temporal and spatial scales, with some processes and states supported by fast frequencies, (e.g., information integration and reasoning), and others by slow ones, [relaxation states, sleep and coma; (Maestú et al., 2015)].

1.4.1 Brain Signals

It is known that when an electric current goes on, a perpendicular and proportional magnetic field can be found associated with it. Neurons, among other mechanisms, communicate through electric currents. The idea, then, is to measure those currents (EEG), or their associated magnetic fields (MEG). These currents/magnetic fields are so weak, that the signal must be cleaned for artifact noises with different origins. For instance, a car at a distance of fifty meters emits a stronger signal that will interfere the measurement. Earth's magnetic field is also a problem, as it is magnitudes stronger than brain signals (Carretie, 2011). To solve this issue, an isolated room is needed to conduct the registers. Signals are so weak, that deep brain areas are inaccessible, and so EEG/MEG only captures superficial information (Maestú et al., 2015). Even more, as information can flow from deep brain regions to cortical ones, the signal obtained is a sum (the exact way currents and fields are joined through brain layers is unknown), and nothing more than the final result, in cortical areas, is captured. It is known that the field decay proportionally to the distance, so the farther the source, the weaker the signal (Maestú, Ríos, & Cabestrero, 2008). Given the anatomical topology of the cortex, a lot of information is lost, and only a proportion of the emitted field is captured. Concretely, only axonal activity, localized in cortex folds can be reached. This is due to the fact that neurons on the folds of the cortex send electric currents that flow parallel to the scalp. As the magnetic field is perpendicular to the current, only those fields will be detected (because the field flows to the outside). Neuronal populations whose current is perpendicular to the scalp (and their magnetic field parallel) won't be detected. *Figure 1.2*, obtained from the Human Connectome web page, shows this idea qualitatively.

As one could expect, in order to detect a so-weak field, very specialized magnetometers are needed, whose maintenance is not easy nor cheap (that is why a MEG machine is that expensive). These sensors are called Superconducting Quantum Interference Devices (SQUIDs), and work under the principle that, when taken almost to the freezing point ($-270°$), some metallic materials amplify extremely their electric/magnetic conductivity (Carretie, 2011). Those materials are called superconductors, and are commonly used in other fields. To maintain such a low temperature, hundreds of litres of liquid helium (the most expensive substance on Earth) are needed. SQUIDs are attached to one or two coils (called single or double coiled SQUIDs), and an electric current is passed through it (them). The current produces a magnetic field, and any perturbation of this field is sensed by the SQUIDs integrated in the circuit. Perturbations may come from many sources (Earth's magnetic field, nearby electrical devices...), but if the sources are isolated, it can be assumed that they have a brain origin (Maestú et al., 2015; Carretie, 2011).

Each sensor, then, is a channel that captures variations on the magnetic field, offering a time series to be filtered and analyzed. This is the way MEG technique captures brain information. The number of sensors depends on the machine's model, and generally vary from 124 up to 512. In our analysis, each sensor or channel will be a variable that produces a time series of observations. Connectivity measures are based on correlations among this series.

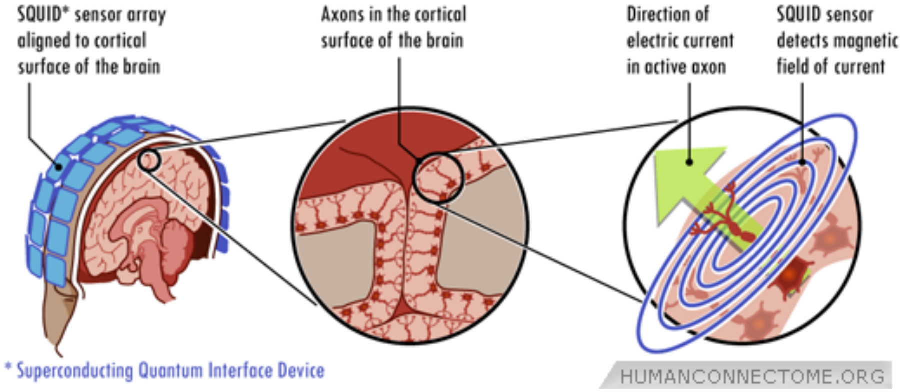

SQUID* sensor array aligned to cortical surface of the brain
Axons in the cortical surface of the brain
Direction of electric current in active axon
SQUID sensor detects magnetic field of current
* Superconducting Quantum Interface Device
HUMANCONNECTOME.ORG

Figure 1.2: SQUID sensors, cortical surface, origins and orientation of the magnetic fields as measured by MEG. Obtained from http://humanconnectome.org/study/hcp-young-adult/project-protocol/meg-eeg

The aforementioned artifact noises have an external origin, and can be avoided filtering the signal. In this sense, isolated rooms acts as a filter. The position and distance of the coils also serves to that purpose, and is not a minor issue, as it allows to cancellate fields of certain frequencies, that precisely depend on the distance of the coils(Maestú et al., 2008). Nevertheless, noise interferences from other sources are common. Head movements, eye blinking and volume conduction are major examples of this interferences, impossible to avoid by physical or analog filters. EEG and MEG suffer from it, but in the case of EEG the problem is more pronounced. This poor conditions have led to the development of sophisticated digital filtering techniques, that aim to maximize the signal-to-noise ratio [SNR, (Carretie, 2011)]. In the process, a lot information is lost. The topic is not trivial, and is a highly technical field leaded by engineers and physics. Roughly speaking, interferences (noise or artifacts) can be divided into two groups (independently of the source of the noise): constant or cyclic and momentary (often modelled as random).

The former is relatively easy to filter, because it is present in all the sample, hopefully through all channels, and occur in a predictable manner. Eye blinking is an example of this type of artifact, and is easily removable (Maestú et al., 2015). Others are not, but even in that case, if the interference affects everything (all the data) in the same way, many measures based on proportional relationships among time series will still offer good results, as proportions are preserved when the noise is constant. Robustness to noise depends on the nature of the measure, as we will see. When the noise is not constant, its presence is not predictable, nor are its effects, and it may be impossible to remove from the data (it can be impossible to distinguish what is signal and what is noise). Examples of this kind of noise are head movements or body adjustment to the machine. Solutions spin around the quest of, at least, a suitable SNR, even though it may imply some loss of information (which is certainly the case).

Yet there is another kind of disturbance that deserves special attention, not necessarily constant if present, from internal origin, and specially problematic: volume conduction. It is a major issue in brain data signal processing, and, hovewer many improvements have been achieved, it's still not resolved. Volume conduction occurs when two or more sensors (generally two) get the same or very similar information not due to coordination phenomena between brain regions. It can be caused because a common, deeper source, is provoking the observed signal in the scalp through two different pathways, each one measured by one sensor, or because sensors are placed between two areas and the received signal is a mixture of the common source and any other one, different for each sensor. Both situations can be found in near sites, and, though hard to solve, the case is relatively easy to detect. When it is due to the location of the sensors (and not from a common source), the perturbance will be present along the entire dataset, thus drawing an implausible coordination profile. As brain synchronization is dynamic and changes over time, it is relatively easy to distinguish and remove. Yet there is another possibility, in which volume conduction is present in very separated sites. It is the less reported case, and certainly the hardest to detect and assess. It is due to the fact that deep brain regions can connect separated cortex neuronal populations through long-range pathways, not necessarily direct (they can be connected via a third involved region), causing the volume conduction problem. Given the lack of a proper brain connectivity model assessing this phenomena, its occurance is unpredictable and highly problematic. Even more, as brain connectivity is inherently dynamic, the issue is not necessarily constant, so detection and removal is almost impossible. The visible part of the quandary is that two sites, separated or not, acquire a very similar signal, which is precisely how synchronization phenomena manifest. In other words, real brain coordination and volume conduction have the same face. As coordination measures look for similarities in the time series, the score will be high when synchronization or volume conduction occur. When volume conduction is detected, smoothing methods that penalty correlation among sites or umbralization are preferential options (Maestú et al., 2015). If the correlation is too high (under a certain criterion), it is weighted or set to zero. This solutions always come at the expense of some loss of information (when the criterion is too conservative), or a worse SNR (when criterion is too relaxed).

All in all, the next question would be, where does this activations come from? where is the source of the observed data? That is the problem of source reconstruction. It is an inverse problem, with no unique solution, very technical and based upon mathematical algorithms (Carretie, 2011). For many scientist, the problem cannot be assesed with today's technology, and solutions are not reliable. For others, the solutions offered by the latests algorithms are, at least, biologically plausible (Maestú et al., 2015). Many models of brain functioning exists, as there is no unified vision of it. Mathematical algorithms that solve the inverse problem are based on those models, thus results vary from one algorithm to another. The mathematical solutions maximize a criteria (extracted from known brain region's communication rules), at the cost of others. Roughly speaking, if the model predicts that brain areas tend to activate together, at the same time, to carry on a function, the algorithm will probably be based on PCA (Principal Components Analysis) and derivatives of it. On the contrary, if it predicts individual and isolated activations, ICA and stem from it will be preferred. In some others prevail energy cost of activation, or information flow efficiency (Carretie, 2011). The methods are also accompanied by brain atlases, and, to tune the model parameters as finer as possible, compared with estimations of explained variance. Examples of this algorithms are (e/s)LORETA, Minimum Norm or Beamforming, to name a few (Maestú et al., 2008; Carretie, 2011).

To sum up, each MEG sensor will get magnetic information associated with electric currents going on in large neuron populations located in folds of the cortex. Many problems of different origins are associated with the measurement, and analog and digital filters are applied in an attempt to solve them. There are also great efforts trying to understand where, and by which pathways, deep structures sends information to the cortex, from which we obtain MEG data.

2 | Objectives and Hypothesis

Synchronization measures are mathematical procedures designed to capture correlations between dynamical processes. Different measures outline different mechanisms of coordination; for example, Phase indexes (PLI, PLV) are based on differences between instantaneous phases of a pair of time series, thus capturing phase couplings. Spectral Coherence is based on the covariances of the spectra of frequencies of time series, and Mutual Information extends the notion of joint entropy to measure coordination as the probability of getting two values simultaneously from two variables. Nowadays, these measures are a standard in neuroscience and signal processing. In the case of brain records, time series come from variations in the hemodynamic response as measured by precession movements of hydrogen spins (Magnetic Resonance), variations in electric currents (EEG) or in their associated magnetic fields [MEG; (Carretie, 2011)], to name a few. The measured source produces a time series, composed of as many variables as sensors, that record a set of values evolving over time. All of the aforementioned synchronization measures try to capture the likeness of those signals or its properties, bivariately or multivariately. Each measure will be described in more detail in the next section.

In this work, the coordination measures considered are Phase Locking Value (PLV), Mutual Information (MI), Spectral Coherence (κ) and Pearson Correlation Coefficient (ρ). The choice comes from the fact that the first three are widely used in the field, and so their properties, advantages and disadvantages are well known. ρ has been chosen due to its ubiquity in statistics, although its use in neuroscience is not so common (more sophisticated measures are preferred over ρ).

Likewise any other statistical procedure, coordination measures work under certain mathematical assumptions. These measures are so well established that more often than not, those assumptions are not checked when analyzing neural time series.

Coherence correlation, a classical measure of coordination, is a linear correlation in the frequency domain, thus relying on the linearity of the spectrum of frequencies. The same can be said of Pearson Correlation Coefficient, applied to the time domain. PLV is calculated as the averaged difference of phases of two signals, so if the distribution of differences is not symmetric, the mean won't be a good representation of it. Even more, extraction of instantaneous phases (and power spectra) to be compared requires the calculation of Fourier Transforms and then Hilbert transforms (or wavelet instead). Both procedures are reliable only if stationarity can be assumed (Cohen, 2014), so PLV, indirectly, and Coherence, directly, depend also on stationarity. Mutual Information is very sensitive to sample size (not a problem in this case) and SNR (signal to noise ratio). In (E)MEG signals, differencing what is signal and what is noise unambiguously is impossible. The only possible solution is an estimation of SNR, calculated as a ratio between average and standard deviation (Cohen, 2014). The properties to be studied, then, are: **linearity**, **normality**, **signal to noise ratio** and **stationarity**. These properties are a basic characterization of any time series and have a direct effect on coordination measures.

Brain signals are measured by sensors, and then filtered to clean the signal from artifacts and noise. Too noisy or faulty segments are removed. Thus, the whole signal is chopped into samples for further analysis. However, it is common practice to re-join them (after sampling) to get a larger time series, expecting to enhance the robustness of the statistical analysis (see below). It is not a different experimental condition, but a methodological necessity. The time gap between samples is unknown, and its temporal continuity is not assured; thus joining all together (concatenated samples) means to combine different fragments, with, possibly, different dynamical regimes and properties. Statistical properties of the signal, like stationarity or linearity, may change when joining all the time series, thus affecting the robustness of the analysis and the validity of results.

Then, the questions to be assessed are: *does the dataset meet the assumptions made by the coordination measures? are there any difference between the joined data and the sampled data? if so, how does the violation of the mathematical assumptions affect coordination measures?*. To answer these questions we'll study both conditions, sampled time series and joined time series. The properties of the signal will be studied in both cases, and the connectivity matrices (coordination between sensors) will be calculated, in an attempt to observe the differences in the results as a function of the properties of the signal and the violation or not of the statistical assumptions. Hence, the TWO MAIN OBJECTIVES OF THIS WORK ARE:

1. ASSESSING THE MATHEMATICAL ASSUMPTIONS THAT UNDERLIE CONNECTIVITY MEASURES, and

2. SEE IF THERE ARE DIFFERENCES WHEN DATA IS SAMPLED AND REJOINED.

3 | Methods

3.1 | Data Collection

In this study, only resting-state MEG data, from one healthy subject, will be analyzed. No task is required during data acquisition, and the participant must be awake. This experimental design is called resting-state, and serves to the purpose of understanding the basal level of functioning of the brain, as opposed to task designs, in which a concrete function or impairment (and thus, concrete brain pathways and regions) are studied.

Neural data has been acquired from The Human Connectome Project's open database (HCP[1]). The project offers open acces to all data, and due to the collection procedure, any identification is impossible. The HCP has a huge goal: map the human connectome as accurately as possible, in normal adults. A connectome is a wiring diagram, a map of the neural conections of any nervous system (Sporns, 2011). The paradigmatic example of a connectome is the one extracted from the *C. Elegans*, a nematode (roundworm) that lengths about 1mm, whose nervous system is completely mapped. In order to discover this map in human beings' brain, neuroscientists from several research centers have been collecting neural data from Magnetic Resonance in two complementary modalities, diffusion imaging and resting-state functional Magnetic Resonance Imaging (fMRI), and from Electroencephalogram (EEG) and Magnetoencephalogram (MEG).

Data were recorded on a whole-head MAGNES 3600 scanner (4D Neuroimaging, San Diego, CA, USA), and belongs to the HCP MEG2 release (Larson-Prior et al., 2013), in which 67 subjects were recorded in resting state. At the end, 61 subjects are available for analysis. The other six recordings didn't pass the quality control checks, that consist in tests for excessive SQUID jumps, correlations between sensors, and sufficiently well-behaved recording channels (Colclough et al., 2016). The room was shielded to avoid magnetic noises, and the scanner is composed of 248 magnetometers and 23 reference channels (5 gradiometer, and 18 magnetometer). In addition, 5 current coils are used in combination with structural imaging data and head surface tracings, in order to localize the brain in relation to the magnetometers. The coils also serve to the purpose of monitoring and correcting head movements. More detailed information can be found in the

[1] http://www.humanconnectome.org/

web page of the project.

Resting-state MEG data are recorded in three consecutive sessions, with little or no break in between, for six minutes each. Then, data is preprocessed with a pipeline that removes artifactual segments of time, identifies faulty channels and applies a zero-phase anti-aliasing filter. The criteria is based on Independent Component Analysis: when a segment is identified as an independent component in the ICA decomposition, with clear artifactual signatures (eye blinks or cardiac interference, for instance), it is removed. The sampling frequency is set to 508.12 Hz, meaning that in each second, each sensor captures 508 values (in the order of the femtotesla in MEG signals). This filtering and removal procedures lead to a data file with several samples of time series recorded by sensors. The exact time gap between samples is unknown. Even though, in order to get a bigger data set and maximize results, it is common practice to join all samples and conduct the analysis.

In this work, we will only analyze the subject 100307, first session, that corresponds to a healthy female between 22 and 35 years. Each sensor or channel is one variable, recording a time series, a set of values or observations over time. Sensors are placed all around the scalp. That means that all brain cortex's surface is recorded at the same time. For the purpose of the analysis, and given that the subject was recorded in resting-state, it does not matter where sensors are concretely placed in relation to brain areas. No analysis on brain regions will be conducted. The analysis will focus on the basal level of brain's activity, and not in concrete areas related to certain tasks.

All dataset (every recorded session) offered by the HCP is divided into samples of equal length. This is a common procedure, derived from artifact removal and criteria established for time series. Every recording is processed to look for problems of volume conduction, magnetic interferences, head movements, etc (as stated before). This means that some fragments of the time series will be removed (precisely to clean it from the aforementioned problems), leading to a set of smaller time series, denominated samples or trials, that are set to be of equal length. It is impossible to assure the continuity of the samples, or the time gap between them. This is a necessary step for reliability, and does not imply any change in the experimental conditions. In any case, it is common practice to re-join all samples after preprocessing in one single time series, to get more data points and enhance analysis. That is the reason to calculate connectivity in both conditions in this study, when the time series is divided into samples, and with the signal taken as a whole, concatenating the samples. The number of samples vary from one subject to another (because sampling is due to artifact removal).

In the considered MEG signals (one subject) for this work, we have *147* matrices (samples), each one composed of *241* channels along with *1018* observations. At a frequency sampling of *508.6 Hz*, each sample comprises about *2* seconds. When joined, the data is one single matrix of *241* channels by 1018×147 samples $= 149646$ observations, around 5 minutes. Joining the data consist simply in putting the samples altogether, one after the other. It is just the same data, concatenated. It is not a different experimental condition, but given the fact that this procedure is common in brain connectivity analysis, the purpose of this work is to asses if this is a reliable approach when calculating synchronization in dynamical brain signals.

3.2 | Building Functional Brain Networks: Synchronization measures

Synchronization (or coordination) measures (SM) are the first step in complex networks analysis of brain signals (and any other discipline dealing with time series). Many SM used in today's neuroscience come from other disciplines like physics and communications, that have been dealing with dynamic systems for a long time. Dynamic systems, like the brain, are characterized by components oscillating over time. Oscillations, in the case of the brain, can be of many natures, from electric currents to biochemical exchange between neurons. When oscillators interact, their dynamics change, and they are said to be coupled. The concrete mechanisms that underlie regions' couplings is unknown, and it is possible that brain coordination works through several different ways, depending on task requirements and energy needs. Its manifestations would be in different characteristic of the signal, like amplitude or phase. Nevertheless, the only known way to observe the brain non-invasively and as whole is through noisy, indirect and highly variable signals obtained from very precise sensors. This unfortunate situation makes interpretation of results and theory building much more difficult, and has boost the implementation of many different coordination measures. There is no agreement on which is the best one, probably due to the fact that, if coordination works under different mechanisms, each SM could be capturing only one manifestation of the process. Experimental results seem to confirm this idea, as some tasks are better defined studying frequencies and others studying phase couplings (Pereda, Quiroga, & Bhattacharya, 2005). SM, then, try to capture how similar two time series are, in an attempt to study how coordinated two brain regions are. Some measures are Model-Based (Granger Causality, for instance), others are free of any model (e.g. Coherence). Most of them assume stationarity, although it may be not the case in biological systems (Stam, 2005). Measures based on the temporal lag of the observed coupling try to capture causal relationships in the form of direction from one region to another (the so-called efective connectivity). In fact, almost any measure can be transformed to be this way introducing a lag in the calculation, thus correlating values of one variable with future values of another, and watching if previous values are usefull to predict future ones (Pereda et al., 2005). Measures can also be bivariate (which is almost always the case), or multivariate. Multivariate approaches are based on the following strategy: given three variables (each one a time series in our case) X,Y,Z calculate the coordination measure for $X - Y$ and $X - Z$. Then normalize by the correlation between $X - Z$ and you get the coordination of X and Y given Z (Pereda et al., 2005).

In this work we will study bivariate undirected measures. Concretely, Pearson Correlation Coefficient, Mutual Information, Coherence and Phase Locking Value. The reason to choose this measure and not others is that we want to compare them in two different conditions (time series in samples and joined time series) whose mathematical properties may change. Then, all of them must be of the same nature to be compared (functional connectivity in this case). Functional connectivity measures are much more common and well studied than effective connectivity measures, computationally much less demanding, and much easier to understand and interpret.

The process will be to calculate the connectivity of one MEG register, in the form of samples and a joined time series. In both conditions, Pearson, Coherence, Mutual Information and PLV will be calculated, reflecting the coordination between a pair of sensors, that are indirectly measuring neuronal populations' magnetic fields. It should be noted that the value of synchronization between any pair of sensors is just a snapshot, or an average, of a dynamical process that evolves over time. Thus, after the calculation of each measure we'll end up with one connectivity matrix per measure for the joined condition, and one connectivity matrix per sample and measure for the sampled condition. Each matrix will be composed of 241 rows by 241 columns, each cell containing the coupling strength between two channels. Once connectivity matrices are calculated, the next step would be to measure network properties such as centrality, degree or clustering. In this work we will only study the properties of the signal and its connectivity through the aforementioned SM.

In the following we present each measure, its basic properties and the feature of the signal its based on.

3.2.1 Pearson's Correlation Coefficient

Measure introduced by Pearson, it is a covariance scaled by variances, thus capturing linear relationships among variables. From the equations of the variance (of X and Y) and covariance (of XY), we obtain Pearson Correlation Coefficient (3.4)

$$S_Y = \sqrt{\frac{\sum (Y_i - \bar{Y})^2}{n-1}} = \sqrt{\frac{\sum y_i^2}{n-1}} \tag{3.1}$$

$$S_X = \sqrt{\frac{\sum (X_i - \bar{X})^2}{n-1}} = \sqrt{\frac{\sum x_i^2}{n-1}} \tag{3.2}$$

$$S_{XY} = E[(X - E[X])(Y - E[Y])] \tag{3.3}$$

$$R_{XY} = \frac{S_{XY}}{S_X S_Y} \tag{3.4}$$

Pearson's correlation is a measure of linear dependence between any pair of variables, and has the great advantage that we don't need to know how the variables are distributed. On the other hand, it is only useful if variables are lineary related to each other. Biological organisms (and the brain and subsequent signals) are considered nonlinear (Wang et al., 2014), dissipative systems (Stam, 2005). Nevertheless, it sometimes offer good results, as stated by some simulation works on network structure recovery by several connectivity measures (Wang et al., 2014).

3.2.2 Coherence

Coherence (magnitude squared coherence or coherence spectrum) is one of the most classical connectivity measures considered. It has some disadvantages that lead to a decay in its use in the field. For example, it doesn't discern the effects of amplitude and phase in the relationships measured between two signals, thus its interpretation is not clear (Lachaux, Rodriguez, Martinerie, & Varela, 1999; Varela, Lachaux, Rodriguez, & Martinerie, 2001). It is a measure of the linear correlation among the two spectra considered, as a function of the frequency (Pereda et al., 2005).

To get the coherence spectrum, data must be in the frequency domain, but, as the data is of finite size, the true spectrum must be estimated. In order to do so, MEG data is usually divided into M sections of equal size (normally around 8). The Fourier Transform (Fast Fourier Transform algorithm) is then computed over the sections, to get the estimate of each section's spectrum (periodogram). Then, the spectra of the sections is averaged to get the estimation of the data, as a whole (Welch's method). Coherence is a normalization of this estimate by the individual autospectral density function (Pereda et al., 2005). It is usually calculated as follows:

$$\widehat{\kappa}^2_{xy}(f) = \frac{|\langle S_{xy}(f)^2\rangle|}{|\langle S_{xx}(f)^2\rangle||\langle S_{yy}(f)^2\rangle|} \tag{3.5}$$

Where S_{xy} is the Cross Power Spectral Density (CSPD) of both signals, S_{xx} and S_{yy} are the Power Spectral Density (PSD) of the segmented signals X and Y taken individually, and $\langle\cdot\rangle$ means average over the M segments.

As it doesn't infer direction (causality), because it is a measure of functional connectivity, the adjacency matrix obtained is symmetric. The measure ranges from 0 to 1, being 1 the maximum correlation for a given frequency, and 0 no correlation at all. It infers linear relationships, so if the relation between frequencies isn't, the results will be misleading. It also has a strong assumption about the data: stationarity, which more often than not isn't the case. The aforementioned chop of the signal into sections sometimes fix this issue, as the data can be locally stationary.

3.2.3 Phase Locking Value

Phase Locking Value (PLV from now on) was first introduced by Lachaux et al. (1999), as a new method to measure synchrony among neural populations. It has, at least, two major advantages over the classical coherence measure: it doesn't require data to be stationary, a condition that can rarely be validated, and has a relatively easy interpretation (in terms of phase coupling). Stationarity means that the mean, variance and structure of frequencies in the data remains constant. In classical coherence it is a strong assumption for accuracy and reliability; in PLV isn't. However, the methods used to extract instantaneous phase, a step needed to calculate PLV, rely on stationarity, so indirectly, PLV can be affected by this condition (Cohen, 2014). PLV specifically quantify phase-relationships, while coherence increases with amplitude covariance, and the relative importance of amplitude and phase covariance in the coherence value is not clear. There is no clear interpretation for the changes in coherence between two neural signals, beyond an obvious indication of interdependency (Lachaux et al., 1999). PLV, on the contrary, requires the distribution of phase differences to follow a Von Mises distribution, the normal analogue in polar space (Cohen, 2014).

To obtain PLV, the signal to be treated has to be decomposed in frequency domain via discrete Fourier transform, and it's instantaneous phases and amplitudes obtained from the frequency spectrum. To achieve this, there are several highly reliable methods, such as Morlet wavelet convolution or Hilbert transform (Pereda et al., 2005; Cohen, 2014). In our work we will utilize the latter.

Phase Locking Value (PLV) is obtained averaging over time t:

$$PLV = \frac{1}{N}\left|\sum_{n=1}^{N} \exp\left(i\theta(t,n)\right)\right| \tag{3.6}$$

Where $\theta(t,n)$ is the (instantaneous) phase difference $\phi_x - \phi_y$, the phases to be compared from the signals captured by the sensors. Comparisons are carried out pairwise (bivariate). When the difference between phases is small (thus θ is almost zero), PLV is close to 1. Otherwise, it is close to zero. This means that when phases are similar (thus the wave have close starting point in time), PLV's score are close to one, reflecting phase coupling processes among regions.

Once we have obtained amplitude and phase, we can compute PLV between any pair of signals, averageing over phase differences ($\theta = \phi_2 - \phi_1$). Amplitude is not used in the analysis. Given PLV is an average, it is measuring how constant differences are between signals via phases. In other words, PLV measures how the relative phase is distributed over the unit circle. If the two signals are phase synchronized, the relative phase will occupy a small portion of the circle and mean phase coherence is high. If the opposite is true, the relative phase spreads out over the entire unit circle and mean phase coherence is very low (Pereda et al., 2005). That is the reason why PLV relies on the Von Mises distribution, that is symmetric and unimodal (Cohen, 2014). If it isn't, the average won't represented well enough the distribution of phase differences.

PLV is very sensitive to volume conduction and common origins (Maestú et al., 2015; Cohen, 2014), and so PLI (phase lag index) or wPLI (weighted phase lag index) are lately recommended by some groups, although they're not free of problems. Some authors recommend to report PLV with an estimation of the signal-to-noise ratio, to get an idea of how biased PLV is (Cohen, 2014).

3.2.4 Mutual Information

Mutual Information (MI) is a measure of shared information between any components of a system, between systems, or any other parameter whose value's probability can be estimated. It is based on Shannon's notion of entropy, which, in a general sense, tries to quantify the amount of information contained in a random variable by means of its estimated probability distribution. Mutual information is a bivariate generalization of this notion: it measures the amount of information *shared* between two random variables by means of its joint distribution, or conversely, the amount of information we can obtain from one random variable observing another. This is analogue to measure the dependence between two random variables (Veyrat-Charvillon & Standaert, 2009). Let X and Y be two random variables with $\{x_1, x_2, ...x_n\}$ and $\{y_1, y_2, ...y_n\}$, n possible values with probabilities $p(x)$ and $p(y)$. The Mutual Information of X relative to Y can be written as follows:

$$I(X \cap Y) = \sum_{x \in X, y \in Y} p(x \cap y) \log_2 \frac{p(x \cap y)}{p(x)p(y)} \tag{3.7}$$

$$I(X \cap Y) = H(X) - H(X|Y) \tag{3.8}$$

Where $p(x \cap y)$ is the probability that X has a value of x *while* Y has a value of y, $H(X)$ is the entropy of X and $H(X|Y)$ is the conditional entropy of X and Y.

One of the major advantages of MI is that it captures linear and non-linear relationships among variables. One disadvantage is that it does not explicitly tell the shape of that distribution (Cohen, 2014). To get the mutual information between two random variables, we first need to estimate their probability density distribution, and although many methods exist to estimate them, histograms is the most common, simple and efficient way (Veyrat-Charvillon & Standaert, 2009; Cohen, 2014; Gierlichs, Batina, Tuyls, & Preneel, 2008). The procedure is simple: set a number of bins, containing a range of values, and count the number of values that fall into each bin. Hence, two issues have to be addressed: the number of bins to be used, and the equality or not, in length, of those bins. In general terms, more bins means more information and vice versa (Veyrat-Charvillon & Standaert, 2009), but it is also true that with noisy signals (as the encountered in MEG data), choosing less bins may have the effect of noise reduction (Cohen, 2014). In practice, this means that several distinct samples can fall into the same bin, which reflects the assumption that they stem from the same datum (Gierlichs et al, 2008). The best performance, then, is achieved by a balance between information and noise. As common practice, bins are equally sized, and the number of bins is determined by the Freedman-Diaconis (F-D) rule, that seems to offer the best balance between preservation of information and noise reduction (Cohen, 2014). The F-D rule was first designed to minimize the difference between the empirical probability distribution and the theoretical one. Denotating the bin size as d, it is determined by:

$$d = 2\frac{IQR(x)}{\sqrt[3]{n}} \tag{3.9}$$

where IQR is the interquartile range of the data, and n is the number of observations in the sample. Given that, for n bins denoted as $b(i)$, the probability is estimated as:

$$\hat{P}[y \in b(i)] = \frac{\sharp b(i)}{q} \tag{3.10}$$

Where $\sharp b(i)$ is the number of observations that fall into bin $b(i)$ and $q = \sum_{i=1}^{n} \sharp b(i)$ is the total number of observations (N, total number of values that comprise the data).

This solution to the density estimation problem is not theoretically justified, but empirically demonstrated to yield acceptable results, leading it to be common practice (Batina et al., 2011). Once we build the histogram of each channel (*eq. 3.9*) and estimate bin's probabilities (*eq. 3.10*) we apply *eq. 3.7* between channels pairwise. With histograms, each channel's density distribution is estimated, and we can compare each value registered by the channel or magnetometer, whose probability is known, to its corresponding pair in all other channels. The corresponding pair is the value taken at the same time by other magnetometers or channels, comparing it one by one. Doing this (comparing one value in one channel to the rest of values in all other channels recorded at the same time, pairwise), we get a value of MI for each comparison, that comprises a matrix of a number of rows equal to the number of sensors, and a number of columns equal to the number of bins. Averaging the values channel-wise, we get the adjacency matrix, squared (of length channels x channels), symmetric (because it is undirected), and weighted (because it is not binary).

Equation 3.7 compares joint probabilities against marginal ones. When two values are independent, the product of their marginal probabilities should equal their joint probability. When not, we can state that there is a relationship among them (not necessarily linear), because the probability of finding those values together is greater than the probability of finding them by chance. Thus, somehow, those brain regions are coupled, working together, although we don't know the way it occurs.

Yet there is another problem to be assesed. MI needs to be normalized to be reliable and understandable. If the compared data are independent, its' joint probabilities will be equal to the product of their marginal ones. That means that when data are not related, MI will offer values near 1. The idea, then, is to substract the expected values of MI that could be obtained just by chance (if the sample were random) to the values of MI obtained. Cohen (2014) states that, although this theoretical offset could be obtained analitically, it is much easier (although computationally much more expensive), to calculate hundreds of thousands of times the value of MI having one of the samples' bins reshuffled. If only one sample is reshuffled, only the joint distribution changes. If we reshuffle many times and then calculate MI, we get the distribution of values of MI expected by chance (that is, if the samples were random). We then substract it to the values of MI obtained at first, and the measure gets normalized. This strategy is called permutation test, and is probably the best option. The idea is to get a distribution of MI values under the null hypothesis. When data is randomly distributed, MI values follow a normal distribution with mean near 0 (Cohen, 2014).

3.3 | Testing the Coordination Measures

Every coordination measure must be tested for statistical significance. However, the test vary from one measure to another. In this section we present the statistical tests for significance conducted for each measure.

Following Cohen (2014) κ and ρ's statistical significance will be checked with t-tests;

one-tailed for κ, as it only takes positive values, and two tailed for ρ, as it ranges from -1 to 1. First, both measures must be t-normalized as

$$t = r\sqrt{\frac{n-2}{1-r^2}} \tag{3.11}$$

being r the considered statistic (κ or ρ) and n the number of observations (1018 in each sample and 149646 in the joined signal). It is remarkable that the native function to assess ρ's significance in MATLAB is based on the normal distribution (thus through a gaussian normalization). When the null hypothesis of normality is not met (as tested by KS test, see below), 3.3 will be used to normalize ρ and κ. Then, to test if it is statistically significant or not, the probability of getting that value or a higher one will be calculated (because the t statistic follows a Student-t distribution).

To normalize MI, permutation testing is the preferred choice, as stated by (Cohen, 2014). Due to computational time, only 5 permutations will be done. The data analyzed is composed of 147 samples, each one comprised by 241 channels capturing a signal of 1018 points. The analysis is conducted over the samples and over the whole joined time series. This means that the sampled data is made up of 147 matrices of 241 rows and 1018 columns. When concatenated, the data is presented as one matrix of 241 rows and $1018 \times 147 = 149646$ columns. A complete permutation test of the recommended number of iterations [500 following Cohen (2014)], only for two channels in the joined condition, lasts $2025.19s/60s = 33.75$ minutes. Each channel must be compared with the rest of them, pairwise. That is $C_{241,2} = 241!/(2!(241-2)!) = 28920$ comparisons $\times 33.75$ minutes each $= 976140$ minutes $/(60 \times 24) = 677.9$ days, almost two years. Only for the joined condition. Double that estimate, and you get the expected time needed to calculate the permutation test for both conditions on one computer. Five iterations will last approximately 6.8 days per condition.

Once the permutation test is completed, its mean is used to get a Z score of MI; that is, transformed from bits to a standard statistical Z-value:

$$Z_I = \frac{I - \mu_{I_p}}{\sigma_{I_p}} \tag{3.12}$$

being Z_I the Z-score of MI, and I the values of MI obtained from eq. 3.7. μ_{I_p} is the mean of the results of MI after permutating the time series and σ_{I_p} its standard deviation.

Given the resources available, the optimal approach to test PLV's statistical significance was calculating a Kruskal-Wallis test or the critical PLV value for each sample at a given *p-value* (Cohen, 2014):

$$PLV_{crit} = \sqrt{\frac{-ln(p)}{n}} \tag{3.13}$$

Any value above PLV_{crit} would be considered statistically significative. Both approaches (Kruskal-Wallis and 3.13) suffer from the same problem. As the number of comparisons increase, the probability of Errors Type I raises. Under these circumstances, we decided to perform a Bonferroni correction to enhance the reliability of our results.

The values of κ, ρ, MI and PLV (contained in the adjacency matrix) will be the weights of the links between nodes (sensors). With it, we could start further analysis on connectivity, to study network properties, such as small-worldness, clustering coefficient, efficiency, etc.

3.4 | Testing the Assumptions

As stated before, to be reliable, synchronization measures require some mathematical assumptions to be reached. In this section we present those assumptions and the way to measure them.

Coherence (κ) and Pearson coefficient (ρ), linear correlation coefficients (the former, in the spectrum of frequencies, the latter on the raw signal), assume linearity. That is, the relationship between values (and, presumably, between brain regions), must be linear to be captured by κ or ρ. In addition, both are specially sensitive to constant noise, if it affects differentially to certain channels. If the noise were present in all channels, and all the sample, proportions would be preserved, and the effect on this coefficients wouldn't be so pronounced. But, if noise is constant along the time series in one channel but not in other, then both measures will over or infra estimate the relationship between data.

κ is computed over frequencies. Frequencies are obtained calculating the Fourier transform, which assumes stationarity. Thus, if the signals are nonstationary, κ will give unreliable results.

On the contrary, Mutual Information does not distinguish the nature of the relationship of the data (and is not so affected by nonstationarity), although it doesn't provide information on what kind of relationship is that. MI is specially sensitive to SNR (signal to noise ratio) and sample size, which is not a problem in this case, given the amount of data. On the contrary, SNR must be a problem, and it is impossible to calculate from empirical data. It can only be estimated. That is the reason why some authors recommend to compute MI over band-filtered time series [band-filtering usually entail better SNRs (Cohen, 2014)].

PLV is highly sensitive to volume conduction, and, as stated before, is indirectly sensitive to stationarity. It is calcuated averaging phase differences, that may be due to a common, deeper source, or to the positioin of sensors (in between two sources). If the conduction is constat over time, PLV will overestimate the synchronization between regions (Maestú et al., 2015). If it is transient, it will be smoothed when averaged. But, if the signal is nonstationary, mean phases won't be representative of the phases population, and its averaged differences won't be a good estimator of the synchronization between regions.

That is, stationarity, linearity and SNR play a role in the reliability of synchronization measures. Consequently, all of them must be tested when calculating coordination between MEG sensors. Additionally, normality will be tested to characterize better the time series.

To test whether the dataset follows a normal distribution or not, Kolmogorov Smirnov test (KS) will be conducted over each channel's signal, for both conditions. KS test is a non-parametric test for univariate normality although it may be used as a goodness-of-fit test for any given distribution. In a certain way, it measures the distance between data's probability distribution and any other given distribution. In the case where two or more variables are checked, Mardia's test for multivariate normality can be more suitable. Given the nature of our data, which, as the literature states, seems to be nonlinear, non-normal and nonstationary (Stam, 2005), KS test seems to be more appropriate, as univariate normality is a more relaxed assumption about data than multivariate normality (which is more unlike to be met).

In statistics and signal processing, stationarity is a common and difficult problem, as it may affect several orders of differences in the processes' equations, an issue that can make statistical inference go wrong.

Time series' stationarity is ussually studied fitting different autoregressive models to data. Autorregresive models try to predict future values of the signal using past values of it, weighted by a coefficient and a stationary (or not) process to be added. Also, depending on the sophistication of the model, an innovation variable usually modelled as white gaussian noise that accounts for the errors of the model can be added. In that sense, autoregressive models are linear regressions of a variable over itself.

From this perspective, time series can be characterized as having or not a trend (an intercept constant in the model's equations), and having or not a unit root (where the coefficient of the past values used to predict the present value of the variable equals one). Then, to test stationary in each channel's signal (both in samples and altoghether), at least two models must be fitted to check unit roots and trends. Following (Amano & Van Norden, 1992), an i10 test will be conducted. It is a paired test, comprising the *Augmented Dickey-Fuller* test (ADF) and the *Kwiatkowski, Phillips, Schmidt, and Shin* (KPSS) test. Both are based on the goodness of fit of different models used to predict the signal.

ADF test a time series for a unit root against a trend-stationary alternative model, augmented with lagged difference terms (past values of the signal). Thus, if the null hypothesis is rejected, a trend-stationary process fits better the data. This doesn't mean explicitly that data is a trend-stationary process, but that a model of such characteristics fits it. It is possible that a nonstationary process fits the data as well. Hence, if the null hypothesis is rejected, both a trend-stationary or a nonstationary process could characterize better the data (than a stationary one). In the case of ADF, the alternative hypothesis is referred to the former, but whereas the latter also fits or not is not accounted. This ambiguity is due to the fact that models of very different nature can arbitrarily fit many time series. That is the reason to check KPSS test too, that precisely tests a nonstationary process model to fit data. ADF test, then, assesses the null hypothesis of a unit root using the model H_0:

$$y_t = c + \delta t + \varphi y_{t-1} + \beta_1 \Delta y_{t-1} + \ldots + \beta_p \Delta y_{t-p} + \varepsilon_t \tag{3.14}$$

where Δ is the difference operator: $\Delta y_t = y_t - y_{t-1}$, p is the model order; that is, the number of steps back used to predict the future value of y ($p = 2$ means that y_{t-1} and y_{t-2} will be used to predict y_t). ε_t is the error term, usually modeled as a Gaussian mean zero variable. This model, as well as AR, MA or ARIMA models are common in signal processing, and are used to characterize stochastic processes. Such models can be linear, non-linear, stationary, non-stationary (with different regimes of stationarity), with/without trends, etc. The idea is to adjust different models of known characteristic equations, to check under which regime could the signal be understood. When $\delta = 0$ the model has no trend component, and when $c = 0$ and $\delta = 0$, the model has no drift nor trend.

The null hypothesis of a unit root is $H_0 : \varphi = 1$. The alternative hypothesis states that $H_1 : \varphi < 1$, under the same equation. When the test rejects the null hypothesis, we assume that the model without unit root characterizes better the data. A unit root process is a data-generating process whose first difference is stationary. This means that if we accept H_0 the time course can be modeled as a stationary process. On the contrary, if we reject H_0 we assume that a trend-stationary or a nonstationary process fits better the data than a stationary one. Note that ADF and KPSS tests are conducted over every channel's time series, so there could be a situation in which some channels record a series coherent with the hypothesis of $\varphi = 1$, and some others coherent with the hypothesis of $\varphi < 1$. The method used to estimate the coefficients in the alternative model is ordinary least squares (OLS).

Unit root processes may sometimes be confused with trend-stationary processes; while they share many properties, they are different in many aspects. In both unit root and trend-stationary processes, the mean can be growing or decreasing over time; however, in the presence of a shock, trend-stationary processes are mean-reverting (i.e. transitory, the time series will converge again towards the growing mean, which was not affected by the shock) while unit-root processes have a permanent impact on the mean [i.e. no convergence over time (Kwiatkowski, Phillips, Schmidt, & Shin, 1992)].

On the other hand, KPSS test evaluates stationarity, but following another approach. It assesses the null hypothesis that a univariate time series is trend stationary against the alternative that it is a nonstationary unit root process. In order to do that, it uses the model:

$$y_t = c_t + \delta t + u_{1t} \tag{3.15}$$

$$c_t = c_{t-1} + u_{2t} \tag{3.16}$$

where δ is the trend coefficient, u_{1t} is a stationary process and u_{2t} is an independent and identically distributed process (i.i.d) with mean 0 and variance σ^2. The null hypothesis is $H_0 : \sigma^2 = 0$, implying that c_t is a constant and acts as the model intercept. If $\sigma^2 > 0$

there is a unit root in c.

ADF and KPSS test different models to fit the dataset. Measuring both may yield to a good characterization of the time series, as the tested conditions are mutually exclusive. The possible regimes a time series could be in are covered by both tests hypothesis. If ADF's H_0 is rejected, data is better characterized by trend-stationary or nonstationary models. If KPSS's H_0 is rejected, a nonstationary model fits data better. So, to accept that the best model to understand the data is a nonstationary one, both null hypothesis should be rejected. If data were better predicted by a trend-stationary model, ADF's H_0 should be rejected, and KPSS's H_0 accepted.

To asses SNR, the only available method for empirical data is the ratio between the average and the dispersion (Cohen, 2014), that will be calculated for every channel in both conditions (samples and joined signal): $SNR = \bar{X}/S$.

The last test is used to evaluate if data adjust to a linear series model or not. To do so, a test based on Barnett and Wolff's (2005) methodology will be used. The test contrasts if a general linear series of the form:

$$X_t = \sum_{j=1} \beta_j \varepsilon_{t-j} + \varepsilon_t \quad (t = 1, 2, ...) \tag{3.17}$$

fits a zero mean, ergodic, discrete time series ($\{X_t\}_{t=1}^{\infty}$), from the bispectrum of the data, obtained from the Fourier transform on the 3rd order moment:

$$\mu(r,s) = E(X_t, X_{t+r}, X_{t+s}) \quad \text{where } r \text{ and } s \text{ are integers} \tag{3.18}$$

The rationale is to generate several hundreds of new samples (500 in our analysis) under the null hypothesis (linear model), a technique called *bootstrap*. The bootstrap procedure is conducted over the phase of the signal, so that a non-parametric estimate of the variance in the third-order moment is obtained. This is done under the assumption of linearity, which is the null hypothesis. The hypothesis is tested calculating the upper and lower percentiles of the bootstrapped population of values minus the third-order moment. When calculated, the test statistic is obtained as the sum of the absolute differences between the observed third-order moment and the most extreme value admissible under the null hypothesis, at a significant level α. To avoid dependent summands in the test, discrepancies in variance, and Error Type I due to multiple comparisons, a second bootstrap procedure is used. When the result of the first estimate is greater than the estimation of the second, the null hypothesis will be rejected. When it happens, a linear model doesn't fit the data, and it is reasonable to assume that, with an probability of $100-\alpha$, there are interactions over the third order; that is, non linear interactions.

Summarizing, in our analysis we will calculate the following tests: Kolmogorov-Smirnov test for univariate normality, Augmented Dickey Fuller test for unit roots in stationary data, Kwiatkowski, Phillips, Schmidt, and Shin test for univariate trend stationary data and the Barnett-Wolff test for linearity. Every test will be conducted over the time series data, in both conditions, when series are joined to form a big dataset of 241 rows by 149646 points, and sampled, as 147 matrices of 241 rows by 1018 points each. Then, also on both conditions, Phase Locking Value, Pearson Correlation Coefficient, Mutual Information and Coherence will be obtained and tested. We will search for differences in the connectivity measures and their statistical significance, and try to interpret them under the violation or not of the mathematical assumptions under which they work. The questions to be answered are: is there any difference in linearity, stationarity and/or normality in both conditions? If so, how does it affect to the results of the connectivity measures from which the network is built?

4 | Results

4.1 | Data Description

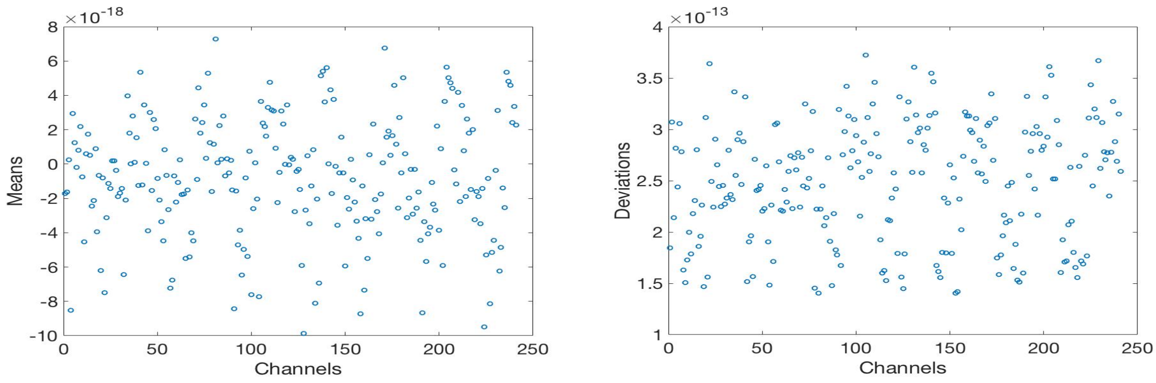
As it is common practice in data-intensive analysis, only the average-cases and extreme values are reported to describe our data.

The channel's average in the join condition equals the global average of samples (mean of the averages of all samples). Almost the same can be said of the standard deviation, where the mean difference of variances between conditions is 3.9×10^{-15}.

Averages among channels, in the joined condition, range from $\bar{X} = -9.87 \times 10^{-18}$ to $\bar{X} = 7.29 \times 10^{-18}$, with a variability of $S^2_{\bar{X}} = 3.44 \times 10^{-18}$, and a global mean of $\bar{X}_{\bar{X}} = -8.01 \times 10^{-18}$. Variability within the channels range from $S^2 = 1.4 \times 10^{-13}$ to $S^2 = 3.72 \times 10^{-13}$, with an average standard deviation of $\bar{X}_{S^2} = 2.48 \times 10^{-13}$. A visual insight can be obtained from plot 4.1 on how means and variances are distributed over channels in the whole time series. Each dot represent a mean (left) or standard deviation (right) in one channel, along the entire time series joined.

Along samples, means range from $\bar{X} = -1.29 \times 10^{-16}$ to $\bar{X} = 1.18 \times 10^{-16}$ (global mean of $\bar{X}_{\bar{X}} = -8.01 \times 10^{-19}$). Variability ranges from $S^2 = 1.01 \times 10^{-13}$ to 6.64×10^{-13} (mean of variances $\bar{X}_{S^2} = 2.44 \times 10^{-13}$). Plot 4.3a (4.3b) shows, along samples, largest averages (standard deviations) (upper dots), medians of averages (std.)(dots in the middle) and smallest averages (std.)(bottom dots). Each row of dots is composed of 147 dots (one per sample), each one representing the largest, median or smallest mean (std.) for one sample. This way, samples are characterized showing the most normal and extreme cases.

It should be noted that samples' means tend to be zero-centered and are two orders of magnitude larger than means when the signal is taken as a whole. Standard deviations are also more variable (have a broader range in the same order of magnitude). That is, joining the time series has the effect of smoothing channels' means and deviations, although globally, the mean is the same in both conditions, and the variance is almost the same. To represent visually this idea, upper (bottom) plot 4.2 shows the smallest and largest average (standard deviation) in samples and in the whole signal, as well as the average mean (standard deviation) shared (almost) in both conditions.



(a) Means in the Joined Condition

(b) St. Dev. in the Joined Condition

Figure 4.1: Each dot represent the mean (left) or standard deviation (right) of one single channel for the joined timeseries.

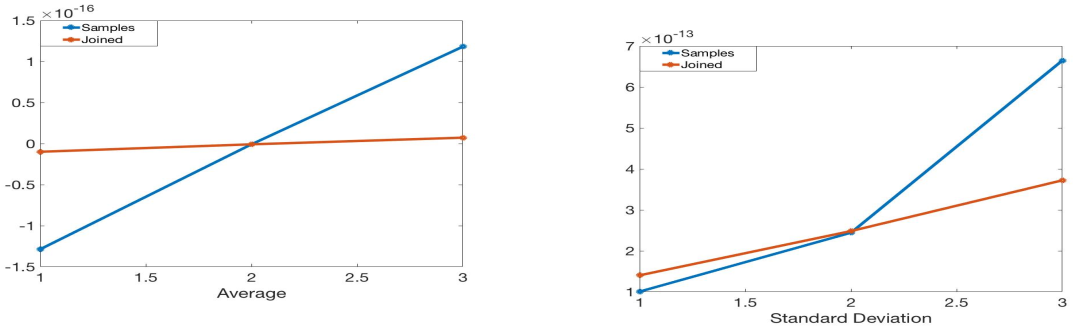


Figure 4.2: For both plots, the red line represents the joined condition, whereas the blue one represents the sampled condition. Left/right plot shows the smallest, most central and largest mean/standard deviation.

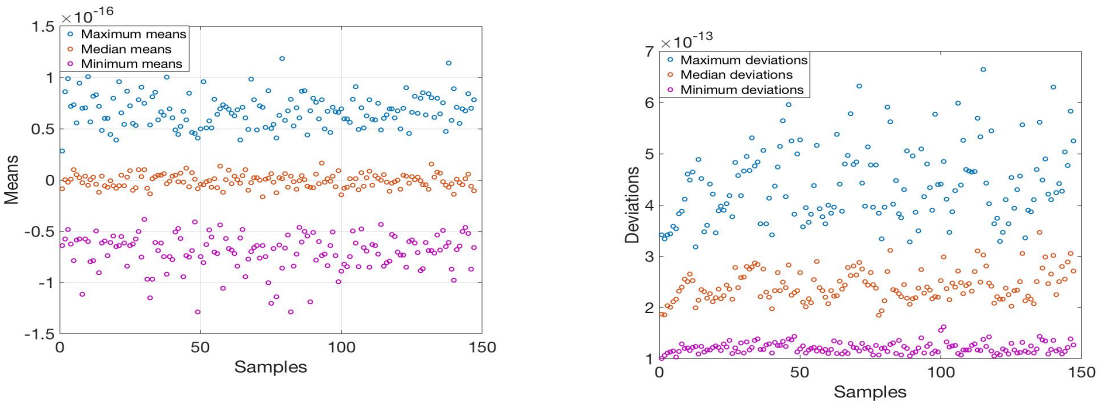


Figure 4.3: Plot of means (left) and deviations (right): all channels (241) along the 147 samples. Upper dots are the largest means/deviations along samples. Middle dots are medians (of means or deviations) along samples and bottom dots are the smallest means/deviations along samples.

4.2 | Results of the assumptions' tests

Kolmogorov Smirnov test for univariate normality: The KS test for both the joined and sampled conditions yielded, a 100% rejections of the null hypothesis. That is, not a single channel's data can be considered as normally distributed on both conditions (in any sample). Note that in all cases, the null hypothesis is rejected with an α always below the corrected $p-value$ (Bonferroni correction). This was an expected results, as it is consistent with the main conception of biological signals.

i10 (ADF and KPSS) test for stationarity:

ADF: When the data is joined, all ADF test's hypothesis are rejected with *p = 0.001*. That is, the null hypothesis of a unit root in the proposed model 3.14 can be rejected for all joined data (for the signals obtained in all 241 channels). That means that we accept the alternative hypothesis, where $\varphi < 1$. The consequences are of great importance in the model, as depending on the value of φ, the system will evolve in a way or in another. As φ grows, the intervals in which y values (observations in the data) go up and down, and the height of those values is increased, and past values of y explain better and more accurately future values of y (Hamilton, 1994). When $|\varphi| \geqslant 1$, covariance-stationary cannot be assumed. In this case, we assume $H_1 = \varphi < 1$, meaning that data, at least in principle, seems to be trend-stationary.

The same can be said in the sampled condition. All time series, in all samples, seem to be adjustable by a trend-stationary process model, as the null hypothesis of unit root is rejected (with $p < 0.001$ for all tests).

That is, ADF test yielded a 100 % of rejections, in both conditions. Consequently, we accept that $\varphi < 1$.

KPSS: KPSS test offered consistent results with ADF. In the joined condition, only in 19 out of 241 cases can the null hypothesis be rejected, all of them reaching a significative *p-value*. 7 of them are between $0.05 > p > 0.04$, 4 are between $0.04 > p > 0.03$, 1 between $0.03 > p > 0.02$, and the other 7 between $0.02 > p > 0.01$. The model in KPSS test (eq. 3.15) considers the hypothesis that the univariate time series is trend stationary, against the alternative that it is a nonstationary unit root process. That is, only for 19 channels can the signal be considered as a nonstationary unit root process. In the other 222 cases, the null hypothesis is not rejected, but $p > 0.05$. Although not statistically significant, the fact that in 222 the null hypothesis could not be rejected is completely coherent with ADF. KPSS's null hypothesis and ADF's alternative hypothesis are analogous. Conversely, KPSS's H_1 is akin to ADF's H_0. Here, analogous means that the proposed models in each hypothesis are of the same nature, trend stationary in ADF's H_1 and KPSS's H_0, and unit root in ADF's H_0 and KPSS's H_1.

Unlike the joined condition, KPSS test's null hypothesis in the sample condition can be rejected in the vast majority of cases. Out of 149646 comparisons (241 channels in 147 samples each), only in 25 cases the null hypothesis is not rejected, and always with $p > 0.05$. There was not a single sample in which the number of not rejections were above 5 cases. Put it in other way, the minimum number of rejections of H_0 is 237, and in 122 out of 147 samples, all 241 channels' time series can be fitted with a nonstationary unit root process. In every case, rejection of the null hypothesis comes with $p < 0.05$. It seems unlikely to be a pattern in the channels in which the null hypothesis has been accepted, although some of them are repeated (never more than twice, and never in adjacent samples).

These results seem to be indicating that the signal, considered in samples, is a non-stationary unit root. But, when joining the time series, this condition is violated, and the process resembles more to a trend stationary model.

Barnett and Wolff's test for Linearity: In the joined condition, 111 out of 241 channels don't fit a linear process ($p < 0.05$). In 25 out of 241, the null hypothesis of linearity must be accepted ($p < 0.05$). That is, no signs of nonlinearity could be found in those channels. Only about half of the signals seem impossible to adjust to a linear model (indicating nonlinear interactions).

On the other hand, in the sample condition, the number of rejections of H_0 (linearity) vary with the sample, between 4 (sample 50) and 133 (sample 81). The mean of rejections is $\bar{X} = 32.048; S^2 = 18.775$. 4802 out of 149646 rejections (about a 3.21%) come with a $p < 0.05$. As shown in figure 4.4, it doesn't seem to be a pattern or change in regime, as could be expected if the experimental design was based on a task, in which presumably, different brain mechanisms of synchronization could lead to different types of relations between time series. The plot shows the number of rejections of H_0 per sample. Keep in mind that the hypothesis is referred to channel's signal, so 241 tests are conducted per sample.

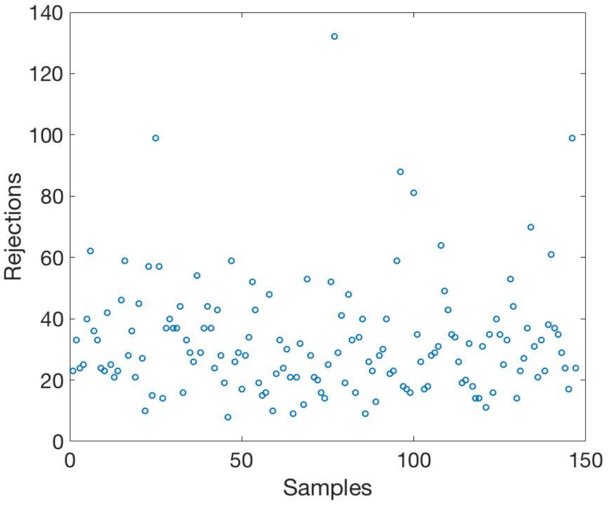


Figure 4.4: Plot of rejections of the null hypothesis of linearity as tested by the Barnett and Wolff method along the 147 samples (with $p < 0.05$).

Signal-to-noise ratio:

SNR is not usually taken into account when analyzing (E/M)EG data. To test its significance, a proper test ad hoc should be created (Cohen, 2014). To do that, non-parametric bootstrap could serve to create a distribution of values under the null hypothesis, but to the author's knowledge there is not a standard procedure to do that. It is out of the scope of this work to create new statistical tests, and values on SNR will be compared informally as the result of dividing the mean to the deviation of the considered signal ($\bar{X}/S$, following Cohen, 2014). In any case, there isn't such a case as a good SNR or a standard acceptable value of it; everything depends on the concrete time series analyzed, by comparison to other time series.

In the joined condition, SNR vary from -3.049×10^{-5} (channel 224) to 3.28×10^{-5} (channel 81). Mean SNR is $\bar{X}_{SNR} = -2.9 \times 10^{-6}$ and standard deviation is $S_{SNR} = 1.28 \times 10^{-5}$. In the sampled condition, SNR vary from -4.77×10^{-4} (channel 45, sample 85), to 4.56×10^{-4} (channel 92, sample 107), with a global mean of $\bar{X}_{SNR} = -2.39 \times 10^{-6}$ and global standard deviation $S_{SNR} = 1.064 \times 10^{-4}$. The correlation coefficient of SNR in the joined signal, and the mean (in channels) along samples is 0.99 ($p \approx 0$), as we can observe in the plot 4.5.

To summarize all this results, KS tests indicates that the hypothesis of normality must be rejected, in all cases and both conditions. Exactly the same can be said of the ADF test (from i10). On the contrary, KPSS yielded a very small amount of rejections (19/241) in

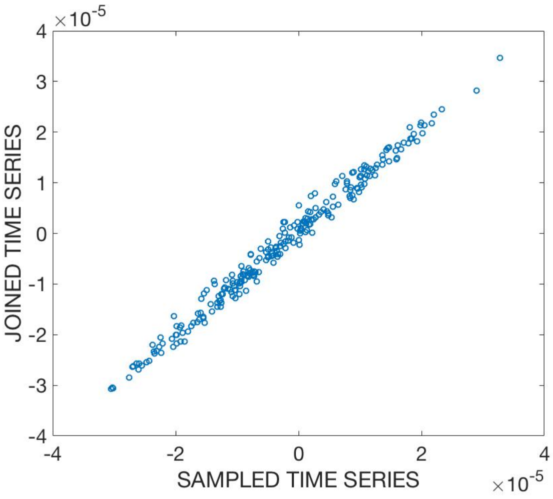


Figure 4.5: Plot of SNR, joined condition vs samples.

one condition (joined time series), but not in the other (samples, 149621/149646 rejections with $p < 0.05$). Barnett and Wolff's test for linearity also yields heterogeneous results: 111/241 rejections of H_0 in the joined condition ($p < 0.05$) and, in average, 32.05/241 significative rejections of H_0 in samples, although only a 3.21% reach $p < 0.05$.

This results seem to indicate that, when the sampled time series are joined together, some mathematical properties of the signal may change. Concretely, the number of rejections of H_0 in KPSS decreases dramatically, indicating that, taken one by one (in samples), this time series could be approximated by a nonstationary root process model (in almost every channel), whereas, when all time series are joined, that is false, and a better approximation is a trend-stationary process model. This means that the regime of both conditions is different, and the behavior of the signal may change from one to another. In presence of a perturbation, a unit root process doesn't converge to the mean over time (the effect is permanent), while trend-stationary processes tend to recover after a transient time. This could mean that, locally, the time series seems to be a nonstationary unit root process, but, if enough time goes on, a more general structure can be grasped, and corresponds to a trend-stationary process.

On the other hand, it is also remarkable that, when taken as a whole, the number of channels that fit to a linear model decreases notably. In other words, taken locally, much more channels can be adjusted by a linear process model. This could be interpreted in a similar way: maybe, when the time series is long enough, it can be seen that it is nonlinear, but taking portions of it, can locally be adjusted by a linear model. This could be tested taking even longer datasets, to see if the rejection of the linear model depends on the length of the considered timeseries.

4.3 | Results of Coordination Measures' tests

As stated before, the best procedure to test the significance of many measures of synchronization is (non-parametric) permutation testing (Cohen, 2014). Due to computational time requirements, this won't be the approach adopted. Instead, equation 3.3 will be used to t-normalize the values of correlation coefficients (Pearson and Coherence), and the probability of getting that value or a higher (κ and ρ) or lower one (for negative values of ρ) will be calculated to test its significance. Mutual Information will be Z-normalized with a 5 iterations permutation test. Permutation test establishes a set of values under the null hypothesis of no relationship between entropy distributions, used later to normalize MI. This procedure corrects for sample size biases and provides a useful framework for addressing multiple-comparisons corrections (Cohen, 2014). PLV will be tested against a threshold critical value of PLV based on a corrected *p-value* (equation 3.13). Table 4.1 summarizes the results found in Pearson Correlation Coefficient, Coherence and PLV. In it, we observe maximum, minimum and average percentage of significative values of the measures across samples, and the percentage of significative values when the time series are joined. The connectivity between any pair of the 241 channels is expressed in an adjacency matrix composed of 241 rows and 241 columns. Given that direction is not inferred, the matrix is symmetric. Thus, the maximum number of significative values is the number of elements in the upper triangle of the matrix minus the number of diagonal elements (that will be zero since the graph has no self-loops), 28920. Note that Mutual Information is obtained normalizing the values generated from the permutation test (with 5 iterations), so it is not included in this table.

	Max (sample)	Mean(std)	Min (sample)	Exact Value (in Joined Cond.)
ρ	92.26 (135)	87.5 (±1.95)	82.1(100)	98.87
κ	82.4 (15)	70.8 (±3.3)	63.8 (3)	100
PLV	88.3 (100)	73.1 (±5.8)	58.9 (78)	99.07

Table 4.1: Maxima, mean and minima percentage of significative values in Pearson Coefficient (ρ), Coherence (κ) and Phase Locking Value (PLV) for samples, and percentage of significative values in the joined time series.

The next table (4.2) explores the range of average and standard deviation values in coordination measures across samples:

	$\bar{X}_{min}$	$\bar{X}$	$\bar{X}_{max}$	S_{min}	S	S_{max}
ρ	0.018	0.042	0.21	0.25	0.308	0.38
$\lvert\rho\rvert$	0.202	0.253	0.35	0.167	0.18	0.21
κ	0.17	0.17	0.19	0.055	0.057	0.068
PLV	0.19	0.24	0.33	0.13	0.14	0.16
MI	0.34	0.39	0.49	0.092	0.13	0.17

Table 4.2: range of Means and Standard Deviations in samples for Pearson Coefficient (ρ) and its absolute value, Coherence (κ), Phase Locking Value (PLV) and Mutual Information (MI)

And table (4.3) displays means and deviations to compare, between the sampled and the joined condition.

	ρ_S	ρ_J	$\lvert\rho_S\rvert$	$\lvert\rho_J\rvert$	κ_S	κ_J	PLV_S	PLV_J	MI_S	MI_J
$\bar{X}$	0.042	0.043	0.25	0.219	0.17	0.169	0.242	0.179	0.393	0.17
S	0.308	0.278	0.18	0.178	0.057	0.055	0.146	0.147	0.127	0.13

Table 4.3: Means and Deviations of Pearson Correlation Coefficient (ρ), Coherence (κ), Phase Locking Value (PLV) and Mutual Information (MI), along samples (S), and in the joined (J) time series.

As shown in table 4.1, the number of significative values for three of the evaluated coordination measures (ρ, κ and PLV) raises notably when the time series is taken as a whole. Keep in mind that, to normalize ρ and κ, eq. 3.3 is used. The parameters needed to compute it are the original value of the coordination measure (r, the value of ρ or κ) and the number of data points (n) used in the analysis, that will vary depending on whether the correlation coefficient works with raw signal (ρ), or its spectrum of frequencies (κ). In samples, for ρ, $n = 1018$ and for κ, $n = 129$. For the whole time series (joined), n of ρ is 149646 (1018×147 samples), and n of κ is 32769. Note that in the normalization of κ, n comes from the frequency spectrum, that is calculated by Welch's method. It works windowing the frequencies obtained from the Fourier transform of the original signal. Once the original values (r) are transformed into t, the probability of getting that value of t, a higher one (one tailed, for κ) or a higher or lower one (two-tailed, for ρ), is calculated from a Student's t-distribution (with a number of degrees of freedom (*d.f.*) equal to

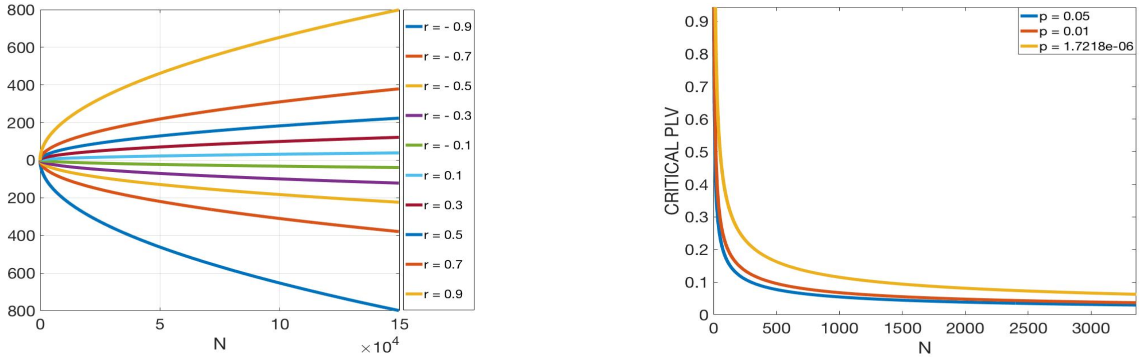
n). When $n \to \infty$, the t-distribution approaches normality. As shown in plot 4.6a, the value of t grows rapidly, thus leading to very unlike values (in the t-distribution) that, subsequently will be evaluated as statistically significative. That seems to be the reason why, for the joined time series, 98.87% of Pearson coefficient, and 100% of Coherence values are considered significative, a 11.37% more in ρ and a 29.2% more in κ.

Something similar happens with the threshold established for PLV, whose percentage of significative values also raises markedly when the time series is taken as a whole (joined). Concretely, it increases a 25.97%. Remember that to obtain a critical value of PLV eq. 3.13 was used. As n enlarges, the value required to classify the original PLV as statistically significative decays exponentially, and stabilizes around 0.009426. The rationale behind this method is that, when the number of instantaneous phases increase, coordination will smooth along the time series. Then, to detect differences, smaller average differences should be considered significative, as the probability of finding them decreases (Cohen, 2014).

In this work, eq. 3.13 has been calculated with $p = 1.7218 \times 10^{-06}$ for the joined condition, and $p = 1.1713 \times 10^{-08}$ for samples, coming from the Bonferroni correction (of $p = 0.05$) for multiple comparisons (29040 comparisons when the signal is joined and 29040×147 samples $= 4268880$ when compared in samples). Plot 4.6b shows how the critical value of PLV decreases as n enlarges, for a fixed original p-value. As it can be observed, the bigger p is, the faster PLV_{CRIT} decays. Conversely, PLV_{CRIT} will be bigger when p is smaller, and for smaller ns and a fixed $p-value$, PLV_{CRIT} will be bigger.

It is also remarkable that, in samples, the percentage of significative values is higher in ρ that in PLV or κ ($\bar{X}_{\%\rho} = 87.5$, $\bar{X}_{\%\kappa} = 70.8$ and $\bar{X}_{\%PLV} = 73.1$), with much less deviation ($S_\rho = 1.95$, $S_\kappa = 3.3$ and $S_{PLV} = 5.8$). That is, systematically, Pearson is classified as significative much more times than PLV or Coherence. The interpretation of this fact is not clear, and will be discussed in more detail in the discussion.

It should be noted as well that the value of ρ tends to zero (0.042 in samples and 0.043 in the signal as a whole). This implies that there is almost as much direct coordination as inverse one. Inverse coordination means that when a brain region is doing something, other region, coordinately, is doing the opposite. Pearson coefficient may not be the best synchronization measure, but it is notable that the other measures, much more common in the field, are blind to this relationships, which, on the other hand, are biologically



(a) Evolution of t as a function of n.

(b) Evolution of PLV_{CRIT} as a function of n.

Figure 4.6: Each line represents the growth of t (left) or *critical value of PLV* (right) as a function of n. In *a)* each line represents the growth of t as n enlarges, keeping r (the original value of ρ or κ) in a fixed value. In *b)* each line represents the decrease in the threshold as n enlarges, keeping a fixed $p-value$. The values tested are $p = 0.05$, $p = 0.01$ and $p = 1.7218 \times 10^{-06}$.

plausible. In any case, to compare the results of all measures at a time, the absolute value of ρ will be used.

As shown in table 4.3, there are only slight differences between both conditions in the values of ρ and κ: ρ differs 0.001 and κ differs 0.001. Standard deviations behave almost the same, in both conditions, for all measures. S_ρ only changes 0.03, S_κ 0.002, S_{PLV} 0.001 and S_{MI} 0.03. On the other hand, PLV changes 0.063 from one condition to another, and MI experiments the biggest change, from $MI = 0.393$ in samples to less than the half, $MI = 0.17$ in the joined condition. This will be discussed in the next section.

Keep in mind that coordination/connectivity analysis is the first step to represent and predict brain activity as a functional network or graph. After this process, network metrics would be extracted. Presumably, the results of it could change due to the calculation of synchronization measures (specially from PLV and Mutual Information). It is out of the scope of this work to go farther, but at first sight, differences in how nodes (sensors) are connected, and the extent to which those nodes are important in the network could change depending on the coordination measure and the condition.

5 | Discussion and Conclusions

In this section, some of the results will be discussed, following the literature in brain connectivity analysis, and some preliminary conclusion will be provided.

There are three major points that worth discussion. First of all, the tests for linearity and stationarity yielded contradictory results, in disagreement with the literature. Second, the significance of PLV, Coherence and Pearson differ notably from one condition to another. Third, the value of Mutual Information decreases considerably depending on the condition.

1. *Linearity and Stationarity*: When data is joined, almost half of the channels' signal (111 out of 241) cannot be adjusted by a linear model. Keep in mind that in this context, the linear model predicts phases (A. G. Barnett & Wolff, 2005). On the contrary, in average only a 13.3% of channels' timeseries cannot be adjusted with a linear model (in samples), and only a 3.21% from that 13.3% reach a confidence level of $p < 0.05$. That is, the sampled signal seems to resemble more a linear process, but the joined signal doesn't. This finding is important in the context of brain network analysis, as generally, brain signals are considered as nonlinear and nonstationary (Stam, 2005). Concerning the latter, other authors argue that, while taken as a whole, MEG and EEG signals are nonstationary, but can be considered as locally stationary (when chopped in samples; Sakkalis, 2011). Even though, the i10 test yielded contradictory results. As shown in the *Results* section (4), the vast majority of channels' signal when the timeseries are sampled can be adjusted better with a unit root nonstationary process model, than with a trend-stationary one, that fits better the signal when joined. This is also a contradictory finding that must be taken into account when analyzing brain signals.

   The reasons for these differences, observable simply by putting samples together, are quite opaque. The time lapse between samples is unknown and arbitrary (as it depends on artifacts and noise present in the timeseries), but, presumably, not very large (signals are recorded in a single session). SNR varies more (one order of magnitude more) when data is joined. This could be indicating that samples are heterogeneous in SNR. This has a lot to do with the quality of the register, presence of noise and even different cognitive states, although resting state is typically considered as random, and differences should cancel out due to the fact that the subject is “not doing anything”. More correctly, is mentally wandering, and not doing a particular task, thus it is considered as a random cognitive processes, which may not be the case if the person focuses his/her attention in different aspects of the outside or inside context. The observed changes in the mathematical properties of the signal could be indicating differences in noise, artifacts (register quality in general), but also in synchronization regimes due to different cognitive activities (that presumably produce different activation and synchronization profiles). If that is the case, there is no reason to combine samples together simply to get a larger dataset, and based upon the supposition that samples belong to the same population (dynamical regime). Joining the time series together implies assuming that every sample accounts the same for the process’ mean and variance, something not proved. That is, samples should be temporally correlated, because they belong to the same process. This is not proved, although resting-state is considered so. The most important conclusion of this work is that, given that these conditions are not necessarily met, and clearly change the mathematical properties of the signal, it is more recommended to conduct connectivity analysis from samples, and then average the results.

These questions could be addressed joining more samples to watch if there is a progression in this trend (as the dataset grows, it resembles more to a nonlinear trend-stationary process), and generating more samples via bootstrap. The procedure could be to generate several thousands of samples for a bigger population of subjects, following different null models, in different dynamical regimes (stationary, nonstationary, trend-stationary). Then, calculating synchronization with the considered measures, we would get a distribution of values under different null models, in both conditions. This would reduce the error and allow better statistical tests to assure when the value of association is statistically significative or not. This proposal could be a future line of research of great importance in MEG research, as it would cover the necessity of proper models to simulate MEG signals and test the significance of connectivity metrics. It would also allow to understand better the subtle relations between the mathematical properties of the signal and the capability of synchronization measures to capture coupling processes. The lack of MEG simulating models is probably due to the prize of a MEG machine, much higher than other brain data signal recorders (EEG, for instance). There exist about 200 machines in the world, and the amount of research from MEG signals is still smaller in comparison with other devices, such as fMRI or DTI. Hence, standards procedures that guarantee robustness and reliability when calculating synchronization measures are still a necessity.

2. *Phase Locking Value, Spectral Coherence and Pearson Coefficient*: As n enlarges to calculate eq. 3.13, the threshold is reduced, and smaller values of PLV are considered statistically significative, under the rationale that in large datasets, coupling smoothes, and smaller differences must be taken into account (Cohen, 2014). Something similar happens to the *t-normalization* of Coherence and Pearson Coefficient (*eqs. 3.3, 3.13*, *figs 4.6a, 4.6b*). Indeed, average values of PLV are smaller for the joined condition than for samples, even though the number of significative values increases from 73.1% up to a 99.07%. In the case of Pearson Coefficient and Coherence, averages are almost the same in both conditions, but significative values increase from 87.5% to 98.87% (Pearson) and 70.8% to 100% (Coherence). This finding is of great importance in brain network science, as the test of statistical significance serves as a threshold for further network calculations. Any value below the threshold would be set to zero; hence, the resulting networks built from one or another condition and synchronization measure will be completely different. The methods proposed to test the significance of this measures (Cohen, 2014) seem to be overestimating the results of synchronization, due to the n. The conclusion, then, is that this methods, at least for large datasets, are not recommended. Instead, it would be preferable to use bootstrap methods, although they are much more time-consuming and computationally much more expensive. It is a limitation in this work not to have implemented bootstrap methods.

3. *Mutual Information*: The average value of MI decreases from 0.39 in samples to 0.17 when the signal is joined, although variance remains between 0.127 and 0.13. As stated before, MI is specially sensitive to SNR. The range of SNR is one order of magnitude bigger in the joined time series, possibly reflecting different regimes of noise and synchronization. It is likely that this condition is affecting the final average of MI, although more research is needed to ensure this. Bootstrap procedures help to avoid this condition stating a distribution of values under the null hypothesis of no relation. It is an important limitation no to have implemented a proper bootstrap procedure (due to resources restrictions). A future line of research, concerning what has been stated before, would be to use the aforementioned null

models (stationary, nonstationary, trend-stationary) no only to account for differences in synchronization values due to the properties of the signal, but controlling also the SNR to observe the differences in MI. Noise, following different probability distributions, can be added easily to artificial signals. Hence, noise could be added to the proposed null models, and correlated with values in MI to understand to which extent is the measure affected by noise.

To sum up, the conclusions are to **conduct connectivity analysis over samples** and not over the whole joined signal, given that the assumptions to do that are not proved and to **avoid normalization for a large *n***, as statistical significance seems to be overestimated (with the equations used in this work). Major limitations in this work are the lack of bootstrap procedures and of simulated MEG signals. And at least, a proposal of a new line of research based on different dynamic process in different regimes of (non)stationarity (with trend or not), to characterize better MEG signals and later simulate them. Each channel would be simulated, and the interactions between them (synchronization) could be known completely, as well as noise. By doing this, a proper model to simulate MEG data could be established, more samples could be bootstrapped to observe if there are changes in MEG signals' mathematical properties on the long-term, statistical significance of coordination measures could be tested more accurately, and the effect of noise in MI would be better understood. At the end, each coordination measure's value could be predicted through a linear regression that takes into account the strength of the coupling between channels, noise, and the parameters of the generative model (trend, unit root or difference step).

Bibliography

Amano, R. A. & Van Norden, S. (1992). *Unit-Root Tests and the Burden of Proof.* Ottawa, Ontario: International Department. Bank of Canada.

Barnett, A. G. & Wolff, R. C. (2005). A Time-Domain Test for Some Types of Nonlinearity. *IEEE Trans. SIGNAL Process. 53*(1). doi:10.1109/TSP.2004.838942

Barnett, L. & Seth, A. K. [Anil K.]. (2011). Behaviour of Granger causality under filtering: Theoretical invariance and practical application. *J. Neurosci. Methods*, *201*(2), 404–419. doi:10.1016/j.jneumeth.2011.08.010

Barnett, L. & Seth, A. K. [Anil K.]. (2014). The MVGC multivariate Granger causality toolbox: A new approach to Granger-causal inference. *J. Neurosci. Methods*, *223*, 50–68. doi:10.1016/j.jneumeth.2013.10.018

Batina, L., Gierlichs, B., Prouff, E., Rivain, M., Standaert, F.-X., & Veyrat-Charvillon, N. (2011). Mutual Information Analysis : a Comprehensive Study. *J. Cryptol. 24*, 269–291. doi:10.1007/s00145-010-9084-8

Bullmore, E. & Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and functional systems. *Nat. Publ. Gr. 10*(3), 186–198. doi:10.1038/nrn2575

Carretie, L. (2011). *Anatomía de la mente: Emoción, cognición y cerebro.* Madrid: Ediciones Pirámide.

Cohen, M. X. (2014). *Analyzing Neural Time Series Data: Theory and Practice.* Cambridge, Massachusetts: MIT Press.

Colclough, G. L., Woolrich, M. W., Tewarie, P. K., Brookes, M. J., Quinn, A. J., & Smith, S. M. (2016). How reliable are MEG resting-state connectivity metrics? *Neuroimage*, *138.* doi:10.1016/j.neuroimage.2016.05.070

Érdi, P. (2008). *Complexity Explained.* Berlin: Springer-Verlag. doi:10.1007/978-3-540-35778-0

Estrada, E. (2011). *The Structure of Complex Networks: Theory and Applications.* Croydon: Oxford University Press.

Friston, K., Moran, R., & Seth, A. K. (2013). Analysing connectivity with Granger causality and dynamic causal modelling. *Curr. Opin. Neurobiol. 23*(2), 172–178. doi:10.1016/j.conb.2012.11.010

Gierlichs, B., Batina, L., Tuyls, P., & Preneel, B. (2008). Mutual Information Analysis A Generic Side-Channel Distinguisher. *Cryptogr. Hardw. Embed. Syst. 2008. Lect. Notes Comput. Sci. 5154*, 426–442.

Hamilton, J. D. (1994). *Time Series Analysis* (Princeton). Chichester, West Sussex.

Kwiatkowski, D., Phillips, P. C., Schmidt, P., & Shin, Y. (1992, October). Testing the null hypothesis of stationarity against the alternative of a unit root. *J. Econom. 54*(1-3), 159–178. doi:10.1016/0304-4076(92)90104-Y

Lachaux, J.-p. P., Rodriguez, E., Martinerie, J., & Varela, F. J. (1999). Measuring Phase Synchrony in Brain Signals. *Hum. Brain Mapp. 8*(4), 194–208. doi:10.1002/(SICI) 1097-0193(1999)8:4<194::AID-HBM4>3.0.CO;2-C

Larson-Prior, L., Oostenveld, R., Della Penna, S., Michalareas, G., Prior, F., Babajani-Feremi, A., . . . Snyder, A. (2013). Adding dynamics to the Human Connectome Project with MEG. *Neuroimage*, *80*.

Maestú, F., Pereda, E., & del Pozo, F. (2015). *Conectividad Funcional y Anatómica en el cerebro humano: Análisis de señales y aplicaciones en ciencias de la salud.* Barcelona: Elsevier España, S.L.U.

Maestú, F., Ríos, M., & Cabestrero, R. (Eds.). (2008). *Neuroimagen. Técnicas y procesos cognitivos.* Barcelona: Elsevier Doyma, S.L. Masson.

Martínez Huartos, J. H. (2015). *Analysing Brain Dynamics by Means of Networks Science* (Doctoral dissertation, Universidad Politécnica de Madrid).

Newman, M. (2004). Analysis of weighted networks. *Phys. Rev. E*, *70*.

Newman, M. (2010). *Networks: An introduction.* Oxford University Press. doi:10.1007/978-3-319-03518-5-8

Papo, D., Martínez, J., Ariza, P., Pineda, J. Á., Boccaletti, S., Pineda-Pardo, J. Á., . . . Buldú, J. M. (2015). Las redes funcionales bajo la perspectiva de la teoría de grafos. In F. Maestú, F. del Pozo, & E. Pereda (Eds.), *Conectividad func. y anatómica en el cereb. hum.* (pp. 81–91). doi:10.1016/B978-84-9022-525-7.00008-0

Pereda, E., Quiroga, R. Q., & Bhattacharya, J. (2005). Nonlinear multivariate analysis of neurophysiological signals. *Prog. Neurobiol. 77*(1-2), 1–37. doi:10.1016/j.pneurobio.2005.10.003

Sakkalis, V. (2011). Review of advanced techniques for the estimation of brain connectivity measured with EEG/MEG. *Comput. Biol. Med. 41*(12), 1110–1117. doi:10.1016/j.compbiomed.2011.06.020

Seth, A. K. [A. K.], Barrett, A. B., & Barnett, L. (2015). Granger Causality Analysis in Neuroscience and Neuroimaging. *J. Neurosci. 35*(8), 3293–3297. doi:10.1523/JNEUROSCI.4399-14.2015

Sporns, O. (2011). *Networks of the brain.* Cambridge, Massachusetts: MIT Press.

Stam, C. J. (2005). Nonlinear dynamical analysis of EEG and MEG: Review of an emerging field. *Clin. Neurophysiol. 116*(10), 2266–2301. doi:10.1016/j.clinph.2005.06.011

Strogatz, S. H. (1994). *Nonlinear Dynamics and Chaos: With Applications to physics, biology, chemistry, and engineering.* Perseus Books. doi:10.1063/1.2807947

Strogatz, S. H. (2001). Exploring complex networks. *Nature*, *410*(6825), 268–276. doi:10.1038/35065725

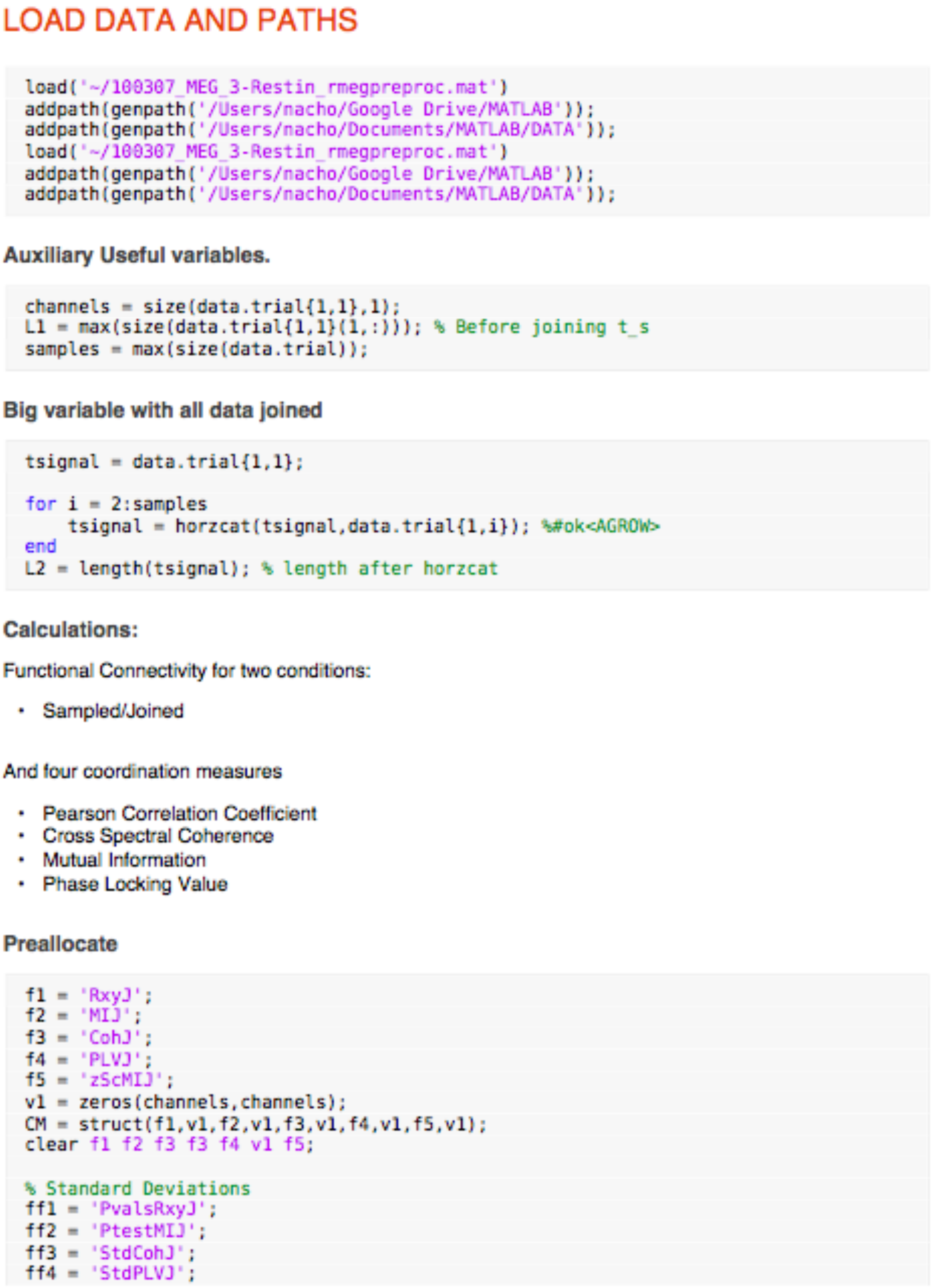
Varela, F., Lachaux, J.-P. P., Rodriguez, E., & Martinerie, J. (2001, April). The brainweb: phase synchronization and large-scale integration. *Nat. Rev. Neurosci. 2*(4), 229–239. doi:10.1038/35067550

Veyrat-Charvillon, N. & Standaert, F.-X. (2009). Mutual Information Analysis : How , When and Why ? *Cryptogr. Hardw. Embed. Syst. 2009. Lect. Notes Comput. Sci. 5747*, 429–443.

Wang, H. E., Bénar, C. G., Pascale, P., Friston, K. J., Jirsa, V. K., & Bernard, C. (2014). A systematic framework for functional connectivity measures. *Front. Neurosci. 8*, 1–22. doi:10.3389/fnins.2014.00405

6 | MATLAB Code

Connectivity



LOAD DATA AND PATHS

```
load('~/100307_MEG_3-Restin_rmegpreproc.mat')
addpath(genpath('/Users/nacho/Google Drive/MATLAB'));
addpath(genpath('/Users/nacho/Documents/MATLAB/DATA'));
load('~/100307_MEG_3-Restin_rmegpreproc.mat')
addpath(genpath('/Users/nacho/Google Drive/MATLAB'));
addpath(genpath('/Users/nacho/Documents/MATLAB/DATA'));
```

Auxiliary Useful variables.

```
channels = size(data.trial{1,1},1);
L1 = max(size(data.trial{1,1}(1,:))); % Before joining t_s
samples = max(size(data.trial));
```

Big variable with all data joined

```
tsignal = data.trial{1,1};

for i = 2:samples
    tsignal = horzcat(tsignal,data.trial{1,i}); %#ok<AGROW>
end
L2 = length(tsignal); % length after horzcat
```

Calculations:

Functional Connectivity for two conditions:

- Sampled/Joined

And four coordination measures

- Pearson Correlation Coefficient
- Cross Spectral Coherence
- Mutual Information
- Phase Locking Value

Preallocate

```
f1 = 'RxyJ';
f2 = 'MIJ';
f3 = 'CohJ';
f4 = 'PLVJ';
f5 = 'zScMIJ';
v1 = zeros(channels,channels);
CM = struct(f1,v1,f2,v1,f3,v1,f4,v1,f5,v1);
clear f1 f2 f3 f3 f4 v1 f5;

% Standard Deviations
ff1 = 'PvalsRxyJ';
ff2 = 'PtestMIJ';
ff3 = 'StdCohJ';
ff4 = 'StdPLVJ';
```

```
vv1 = zeros(channels,channels);
Std = struct(ff1,vv1,ff2,vv1,ff3,vv1,ff4,vv1);
clear ff1 ff2 ff3 ff3 ff4 vv1;

field1 = 'Averages';
v2 = CM;
field2 = 'Std';
v3 = Std;
JC = struct(field1,v2,field2,v3); % Joined Connectivity
clear field1 field2 v2 v3 CM Std;
```

Calculate measures:

Joined Samples Coherence and PLV:

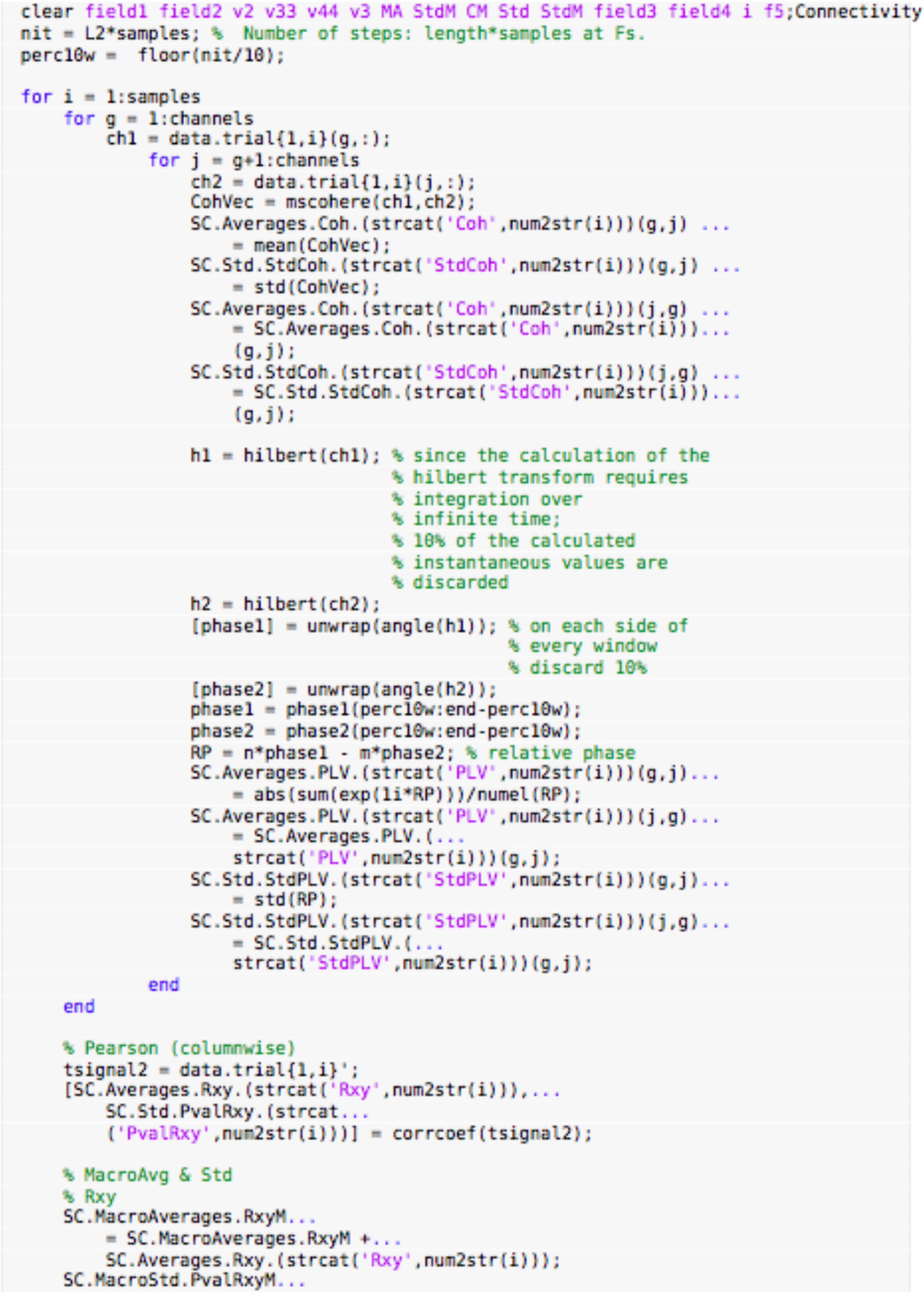
```
% Sync Indices (smilar to locking constant
n = 1;
% Set to 1 (it's the same system, everything coming from MEG.
% Otherwise we would be assuming some interaction between signals
% whose existance is not proved)
m = 1;
%  Number of steps: length*samples at Fs.
nit = L1*samples;
perc10w =  floor(nit/10);

for g = 1:channels
    ch1 = tsignal(g,:);
        for j = g+1:channels
            ch2 = tsignal(j,:);
            CohVec = mscohere(ch1,ch2);
            JC.Averages.CohJ(g,j) = mean(CohVec);
            JC.Std.StdCohJ(g,j) = std(CohVec);
            JC.Averages.CohJ(j,g) = JC.Averages.CohJ(g,j);
            JC.Std.StdCohJ(j,g) = JC.Std.StdCohJ(g,j);

            h1 = hilbert(ch1); % since the calculation of the
                               % hilbert transform requires
                               % integration over
                               % infinite time;
                               % 10% of the calculated
                               % instantaneous values are discarded
             h2 = hilbert(ch2);
            [phase1] = unwrap(angle(h1)); % on each side of every
                                          % window, discard 10%
            [phase2] = unwrap(angle(h2));
            phase1 = phase1(perc10w:end-perc10w);
            phase2 = phase2(perc10w:end-perc10w);
            RP = n*phase1 - m*phase2; % relative phase
            JC.Averages.PLVJ(g,j) = abs(sum(exp(1i*RP)))/numel(RP);
            JC.Averages.PLVJ(j,g) = JC.Averages.PLVJ(g,j);
            JC.Std.StdPLVJ(g,j) = std(RP);
            JC.Std.StdPLVJ(j,g) = JC.Std.StdPLVJ(g,j);

        end
end
```

Pearson (columnwise):

```
tsignal2 = tsignal';

[JC.Averages.RxyJ,JC.Std.PvalsRxyJ] = corrcoef(tsignal2);

% save intermediate results
save('JoinnedConnectivity.mat','JC');
```

Sampled Averages

```
for i = 1:samples

    CM.Rxy.(strcat('Rxy',num2str(i))) = zeros(channels);
    CM.PLV.(strcat('PLV',num2str(i))) = zeros(channels);
    CM.Coh.(strcat('Coh',num2str(i))) = zeros(channels);
    CM.MI.(strcat('MI',num2str(i))) = zeros(channels);
    CM.zScMI.(strcat('zScMI',num2str(i))) = zeros(channels);

    Std.PvalRxy.(strcat('PvalRxy',num2str(i)))= zeros(channels);
    Std.StdPLV.(strcat('StdPLV',num2str(i)))=zeros(channels);
    Std.PtestMI.(strcat('PtestMI',num2str(i)))=zeros(channels);
    Std.StdCoh.(strcat('StdCoh',num2str(i)))=zeros(channels);

end

field1 = 'Averages';
v2 = CM;
field2 = 'Std';
v3 = Std;

f1 = 'RxyM';
f2 = 'MIM';
f3 = 'CohM';
f4 = 'PLVM';
f5 = 'zScMIM';
v1 = zeros(channels,channels);
MA = struct(f1,v1,f2,v1,f3,v1,f4,v1,f5,v1);
clear f1 f2 f3 f3 f4 v1;

% Standard Deviations
ff1 = 'PvalRxyM';
ff2 = 'PtestMIM';
ff3 = 'StdCohM';
ff4 = 'StdPLVM';
vv1 = zeros(channels,channels);
StdM = struct(ff1,vv1,ff2,vv1,ff3,vv1,ff4,vv1);
clear ff1 ff2 ff3 ff3 ff4 vv1;

% Samples Averages and Std
field3 = 'MacroAverages';
v33 = MA;
field4 = 'MacroStd';
v44 = StdM;
SC = struct(field1,v2,field2,v3,field3,v33,field4,v44); % Sampled Connectivity
```

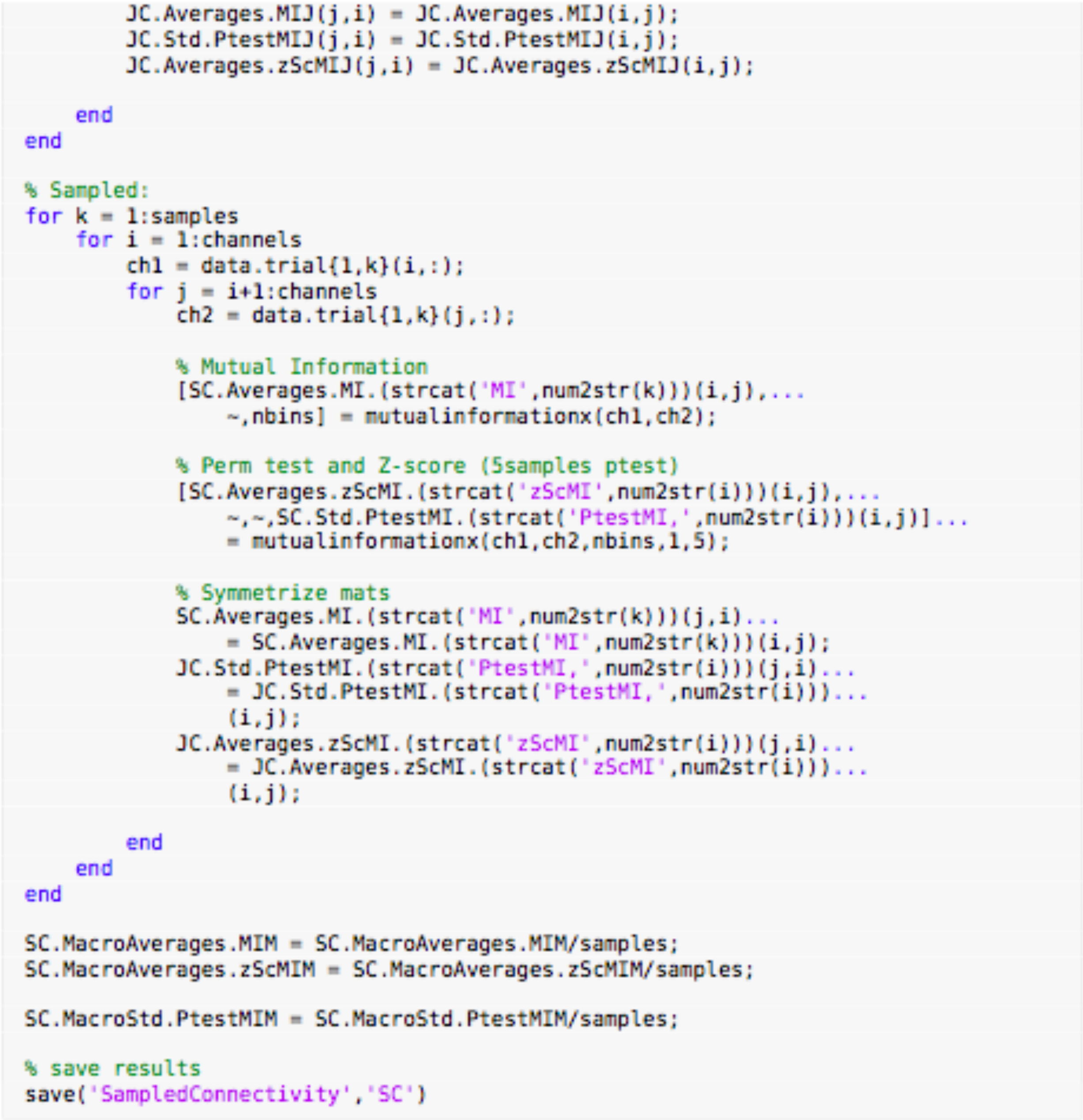


```
clear field1 field2 v2 v33 v44 v3 MA StdM CM Std StdM field3 field4 i f5;Connectivity
nit = L2*samples; %  Number of steps: length*samples at Fs.
perc10w =  floor(nit/10);

for i = 1:samples
    for g = 1:channels
        ch1 = data.trial{1,i}(g,:);
            for j = g+1:channels
                ch2 = data.trial{1,i}(j,:);
                CohVec = mscohere(ch1,ch2);
                SC.Averages.Coh.(strcat('Coh',num2str(i)))(g,j) ...
                    = mean(CohVec);
                SC.Std.StdCoh.(strcat('StdCoh',num2str(i)))(g,j) ...
                    = std(CohVec);
                SC.Averages.Coh.(strcat('Coh',num2str(i)))(j,g) ...
                    = SC.Averages.Coh.(strcat('Coh',num2str(i)))...
                    (g,j);
                SC.Std.StdCoh.(strcat('StdCoh',num2str(i)))(j,g) ...
                    = SC.Std.StdCoh.(strcat('StdCoh',num2str(i)))...
                    (g,j);

                h1 = hilbert(ch1); % since the calculation of the
                                   % hilbert transform requires
                                   % integration over
                                   % infinite time;
                                   % 10% of the calculated
                                   % instantaneous values are
                                   % discarded
                h2 = hilbert(ch2);
                [phase1] = unwrap(angle(h1)); % on each side of
                                              % every window
                                              % discard 10%
                [phase2] = unwrap(angle(h2));
                phase1 = phase1(perc10w:end-perc10w);
                phase2 = phase2(perc10w:end-perc10w);
                RP = n*phase1 - m*phase2; % relative phase
                SC.Averages.PLV.(strcat('PLV',num2str(i)))(g,j)...
                    = abs(sum(exp(1i*RP)))/numel(RP);
                SC.Averages.PLV.(strcat('PLV',num2str(i)))(j,g)...
                    = SC.Averages.PLV.(...
                    strcat('PLV',num2str(i)))(g,j);
                SC.Std.StdPLV.(strcat('StdPLV',num2str(i)))(g,j)...
                    = std(RP);
                SC.Std.StdPLV.(strcat('StdPLV',num2str(i)))(j,g)...
                    = SC.Std.StdPLV.(...
                    strcat('StdPLV',num2str(i)))(g,j);
            end
    end

    % Pearson (columnwise)
    tsignal2 = data.trial{1,i}';
    [SC.Averages.Rxy.(strcat('Rxy',num2str(i))),...
        SC.Std.PvalRxy.(strcat...
        ('PvalRxy',num2str(i)))] = corrcoef(tsignal2);

    % MacroAvg & Std
    % Rxy
    SC.MacroAverages.RxyM...
        = SC.MacroAverages.RxyM +...
        SC.Averages.Rxy.(strcat('Rxy',num2str(i)));
    SC.MacroStd.PvalRxyM...
```

```
            = SC.MacroStd.PvalRxy +...
            SC.Std.PvalRxy.(strcat('PvalRxy',num2str(i)));
        % PLV
        SC.MacroAverages.PLVM...
            = SC.MacroAverages.PLVM +...
            SC.Averages.PLV.(strcat('PLV',num2str(i)));
        SC.MacroStd.StdPLVM...
            = SC.MacroStd.StdPLVM +...
            SC.Std.StdPLV.(strcat('StdPLV',num2str(i)));
        % Coh
        SC.MacroAverages.CohM...
            = SC.MacroAverages.CohM  +...
            SC.Averages.Coh.(strcat('Coh',num2str(i)));
        SC.MacroStd.StdCohM...
            = SC.MacroStd.StdCohM +...
            SC.Std.StdCoh.(strcat('StdCoh',num2str(i)));
end

SC.MacroAverages.RxyM = SC.MacroAverages.RxyM/samples;
SC.MacroAverages.CohM = SC.MacroAverages.CohM/samples;
SC.MacroAverages.PLVM = SC.MacroAverages.PLVM/samples;

SC.MacroStd.PvalRxyM = SC.MacroStd.PvalRxyM/samples;
SC.MacroStd.StdCohM = SC.MacroStd.StdCohM/samples;
SC.MacroStd.StdPLVM = SC.MacroStd.StdPLVM/samples;

% save intermediate results
save('SampledConnectivity','SC')

ch = 1;
for g = 1:ch
    ch1 = tsignal(g,:);
        for j = g+1:ch
            ch2 = tsignal(j,:);
            h1 = hilbert(ch1);
            h2 = hilbert(ch2);
            [phase1] = unwrap(angle(h1));
            [phase2] = unwrap(angle(h2));
            phase1 = phase1(perc10w:end-perc10w);
            phase2 = phase2(perc10w:end-perc10w);
            RP = n*phase1 - m*phase2;
        end
end
```

Mutual Information PermTest

```
for i = 1:channels
    ch1 = tsignal(i,:);
    for j = i+1:channels
        ch2 = tsignal(j,:);

        % Mutual Information
        [JC.Averages.MIJ(i,j),~,nbins]...
            = mutualinformationx(ch1,ch2);

        % Perm test and Z-score (5samples ptest)
        [JC.Averages.zScMIJ(i,j),~,~,JC.Std.PtestMIJ(i,j)]...
            = mutualinformationx(ch1,ch2,nbins,1,5);

        % Symmetrize mats
```



```
        JC.Averages.MIJ(j,i) = JC.Averages.MIJ(i,j);
        JC.Std.PtestMIJ(j,i) = JC.Std.PtestMIJ(i,j);
        JC.Averages.zScMIJ(j,i) = JC.Averages.zScMIJ(i,j);

    end
end

% Sampled:
for k = 1:samples
    for i = 1:channels
        ch1 = data.trial{1,k}(i,:);
        for j = i+1:channels
            ch2 = data.trial{1,k}(j,:);

            % Mutual Information
            [SC.Averages.MI.(strcat('MI',num2str(k)))(i,j),...
                ~,nbins] = mutualinformationx(ch1,ch2);

            % Perm test and Z-score (5samples ptest)
            [SC.Averages.zScMI.(strcat('zScMI',num2str(i)))(i,j),...
                ~,~,SC.Std.PtestMI.(strcat('PtestMI,',num2str(i)))(i,j)]...
                = mutualinformationx(ch1,ch2,nbins,1,5);

            % Symmetrize mats
            SC.Averages.MI.(strcat('MI',num2str(k)))(j,i)...
                = SC.Averages.MI.(strcat('MI',num2str(k)))(i,j);
            JC.Std.PtestMI.(strcat('PtestMI,',num2str(i)))(j,i)...
                = JC.Std.PtestMI.(strcat('PtestMI,',num2str(i)))...
                (i,j);
            JC.Averages.zScMI.(strcat('zScMI',num2str(i)))(j,i)...
                = JC.Averages.zScMI.(strcat('zScMI',num2str(i)))...
                (i,j);

        end
    end
end

SC.MacroAverages.MIM = SC.MacroAverages.MIM/samples;
SC.MacroAverages.zScMIM = SC.MacroAverages.zScMIM/samples;

SC.MacroStd.PtestMIM = SC.MacroStd.PtestMIM/samples;

% save results
save('SampledConnectivity','SC')
```

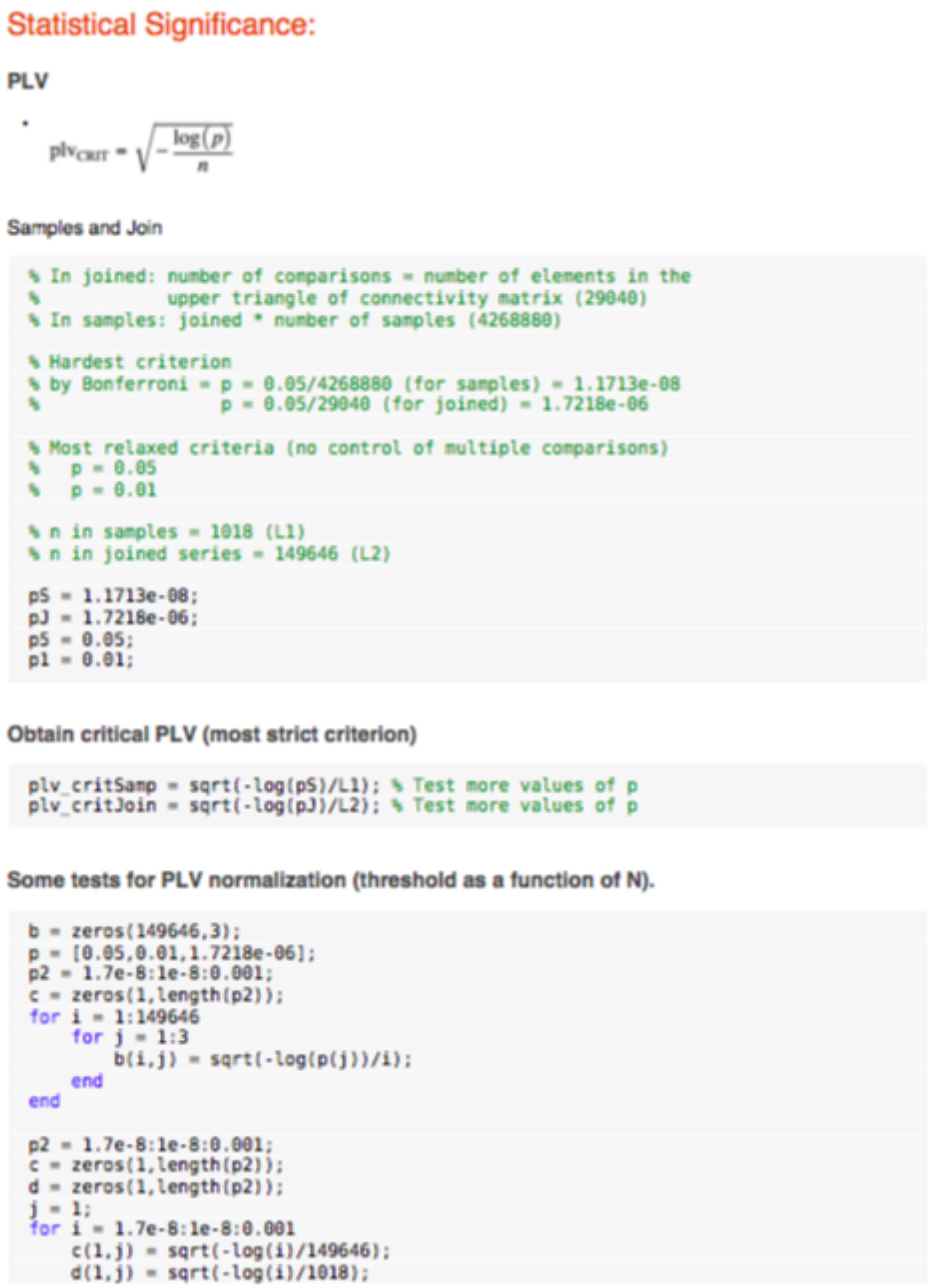
Descriptives

Calculate Descriptive measures

```matlab
Dscps.Global = struct;
Dscps.Global.Samps = zeros(samples,4);
Dscps.Global.Chans = zeros(channels,4);
Dscps.Global.Join = zeros(channels,2);
Dscps.Local = struct;

for i = 1:samples
    Dscps.Local.(strcat('s',num2str(i))) = zeros(channels,2);

    Dscps.Local.(strcat('s',num2str(i)))(:,1) = mean(data.trial{1,i},2);
    Dscps.Local.(strcat('s',num2str(i)))(:,2) = std(data.trial{1,i},0,2);
end

Dscps.Global.Join(:,1) = mean(tsignal);
Dscps.Global.Join(:,2) = std(tsignal);

SampAvgs = zeros(channels,samples);
SampStds = zeros(channels,samples);

for i = 1:samples
    for j = 1:channels
        SampAvgs(:,i) = Dscps.Local.(strcat('s',num2str(i)))(:,1);
        SampStds(:,i) = Dscps.Local.(strcat('s',num2str(i)))(:,2);
    end
end

Dscps.Global.Chans(:,1) = mean(SampAvgs,2); % Mean of channels along samples
Dscps.Global.Chans(:,2) = mean(SampStds,2); % Average Stds of chans along samples
Dscps.Global.Chans(:,3) = std(SampAvgs,0,2); % Deviations of the mean along samples
Dscps.Global.Chans(:,4) = std(SampStds,0,2); % Deviations of Stds along samples

Dscps.Global.Samps(:,1) = mean(SampAvgs); % Mean of samples along channels
Dscps.Global.Samps(:,2) = mean(SampStds); % Average Stds of samples along channels
Dscps.Global.Samps(:,3) = std(SampAvgs); % Deviations of the mean along channels
Dscps.Global.Samps(:,4) = std(SampStds); % Deviations of Stds along channels

% Average differences in Std between conditions
DifStdCond = mean(Dscps.Global.Join(:,2) - Dscps.Global.Chans(:,2));

% clean
close all;

for i = 1:samples
 plot(Dscps.Local.(strcat('s',num2str(i)))(:,2)); hold on;
end

ms = max(SampAvgs);
maxSamp = max(max(SampAvgs));
plot(ms,'o')
mins = min(SampAvgs);
hold on;
plot(mins,'o')
meansSamp = mean(SampAvgs); plot(meansSamp,'o')

figure;
plot(max(SampAvgs),'o')
```

```
hold on;
plot(median(SampAvgs),'o');
plot(min(SampAvgs),'o');

legend('Maximum means','Median means','Minimum means');
ylabel('Means');
xlabel('Samples');

figure;
plot(max(SampStds),'o')
hold on;
plot(median(SampStds),'o');
plot(min(SampStds),'o');
legend('Maximum deviations','Median deviations','Minimum deviations');

boxplot(tsignal(:,128));

tsmeans = mean(tsignal);
tsdevs = std(tsignal);

plot(tsmeans,'o');
plot(tsdevs,'o');

SampAvgs = SampAvgs';
SampStds =SampStds';
close all

a = [min(min(SampAvgs)),mean(mean(SampAvgs)),max(max(SampAvgs))];
b = [min(mean(tsignal)),mean(mean(tsignal)),max(mean(tsignal))];

plot(a,'-o');hold on; plot(b,'-o');
legend('Samples','Join');

c = [min(min(SampStds)),mean(mean(SampStds)),max(max(SampStds))];
d = [min(std(tsignal)),mean(std(tsignal)),max(std(tsignal))];

plot(c,'-o');hold on; plot(d,'-o');
legend('Samples','Join');
```

Significance



Statistical Significance:

PLV

- $$\mathrm{plv}_{\mathrm{CRIT}} = \sqrt{-\frac{\log(p)}{n}}$$

Samples and Join

```
% In joined: number of comparisons = number of elements in the
%            upper triangle of connectivity matrix (29040)
% In samples: joined * number of samples (4268880)

% Hardest criterion
% by Bonferroni = p = 0.05/4268880 (for samples) = 1.1713e-08
%                 p = 0.05/29040 (for joined) = 1.7218e-06

% Most relaxed criteria (no control of multiple comparisons)
%   p = 0.05
%   p = 0.01

% n in samples = 1018 (L1)
% n in joined series = 149646 (L2)

pS = 1.1713e-08;
pJ = 1.7218e-06;
p5 = 0.05;
p1 = 0.01;
```

Obtain critical PLV (most strict criterion)

```
plv_critSamp = sqrt(-log(pS)/L1); % Test more values of p
plv_critJoin = sqrt(-log(pJ)/L2); % Test more values of p
```

Some tests for PLV normalization (threshold as a function of N).

```
b = zeros(149646,3);
p = [0.05,0.01,1.7218e-06];
p2 = 1.7e-8:1e-8:0.001;
c = zeros(1,length(p2));
for i = 1:149646
    for j = 1:3
        b(i,j) = sqrt(-log(p(j))/i);
    end
end

p2 = 1.7e-8:1e-8:0.001;
c = zeros(1,length(p2));
d = zeros(1,length(p2));
j = 1;
for i = 1.7e-8:1e-8:0.001
    c(1,j) = sqrt(-log(i)/149646);
    d(1,j) = sqrt(-log(i)/1018);
    j = j + 1;
end
```

Compute number of statistically significative PLV values

(PLV values greater than threshold)

Samples

```
Max = 241^2; % max number of significative values
PLVSampSig  = zeros(147,1); % Percentage
% With p = 1.1713e-08
for i = 1:147
    PLVSampSig(i) = ...
        ( ( numel( find( SC.Averages.PLV.( strcat( 'PLV',num2str( i ) ) )...
        > plv_critSamp ) ) ) / Max ) * 100;
end
```

New Matrix with Sync Values after Thresholding

```
avgPLVsamp = zeros(channels);
ThreshPLVSamp = cell(147,1);
for i = 1:147 %Samples
    ThreshPLVSamp{i,1} = zeros(channels);
    for ii = 1:241 %CHannels
        for jj = ii+1:241
            if SC.Averages.PLV.( strcat( 'PLV',num2str( i ) ) )(ii,jj)...
                    > plv_critSamp
                ThreshPLVSamp{i,1}(ii,jj)...
                    = SC.Averages.PLV.( strcat( 'PLV',num2str( i ) ) )(ii,jj);
            else
                ThreshPLVSamp{i,1}(ii,jj) = 0;
            end
        end
    end
end

for i = 1:147 %samples
    for ii = 1:channels
        for jj = 1+ii : channels
            ThreshPLVSamp{i,1}(jj,ii) = ThreshPLVSamp{i,1}(ii,jj); %Symmetric
        end
    end
    avgPLVsamp = avgPLVsamp + ThreshPLVSamp{i,1};
end
avgPLVsamp = avgPLVsamp./samples; %Average

% save threshold intermediate values to alleviate computation
save('~/PreResults/Finale/ThreshPLV.mat','ThreshPLVSamp','avgPLVsamp');
```

Joined condition:

Number of sig PLV Over Tresh2 (0.0987, with $\alpha = 4.9116e-05$) (Hardest Criterium)

```
PLVJoinSig = (numel(find(JC.Averages.PLVJ > plv_critJoin))/Max)*100;
```

```
avgPLVjoin = zeros(channels);
for i = 1:channels
    for j = i+1:channels
        if JC.Averages.PLVJ(i,j) > plv_critJoin
            avgPLVjoin(i,j) = JC.Averages.PLVJ(i,j);
        else
            avgPLVjoin(i,j) = 0;
        end
        avgPLVjoin(j,i) = avgPLVjoin(i,j);
    end

end
% save intermediate results
save('/Users/nacho/Google Drive/TFM/PreResults/Finale/ThreshPLVjoin.mat','avgPLVjoin')
```

Coherence and Pearson: Following Cohen(2014):

To test the statistical significance of correlation coefficients:

- $t = r \cdot \sqrt{\frac{n-1}{1-r^2}}$

r is the correlation coefficient, n number of data points in Pearson and number of data in Spectra (Coherence) n-2 d.f

- n = L2 (join, Pearson)

or

- L1 (samples, Pearson) n = 32769 (join, Coherence)

or

- 129 (samples, Coherence)

Join

```
tJoinRho = zeros(241);
tJoinRhoPval = zeros(241);
tJoinCoh = zeros(241);
tJoinCohPval = zeros(241);
n = 32769;

for i = 1:channels
    for j = i+1:channels

        r = JC.Averages.RxyJ(i,j);
        t = r * sqrt( (L2 - 1) / (1 - r^2) );

        tJoinRho(i,j) = t;
            tJoinRho(j,i) = tJoinRho(i,j);  % Symmetric
        tJoinRhoPval(i,j) = 1-tcdf(abs(t),L2-2);
```

```
         tJoinRhoPval(j,i) = tJoinRhoPval(i,j); % Symmetric

        r = JC.Averages.CohJ(i,j);
        t = r * sqrt( (n - 1) / (1 - r^2) );

        tJoinCoh(i,j) = t;
            tJoinCoh(j,i) = tJoinCoh(i,j);  % Symmetric
        tJoinCohPval(i,j) = 1-tcdf(abs(t),n-2);
         tJoinCohPval(j,i) = tJoinCohPval(i,j); % Symmetric

    end
end

pcgRhoJoin = (numel(find(tJoinRhoPval<= 0.05))/(Max))*100;
pcgCohJoin = (numel(find(tJoinCohPval<= 0.05))/(Max))*100;
```

Threshold

```
RhoThreshJoin = zeros(channels);
CohThreshJoin = zeros(channels);
for i = 1:channels
    for j = 1+i:channels
        if tJoinCohPval(i,j) <= 0.001
            CohThreshJoin(i,j) = tJoinCoh(i,j);
        else
            CohThreshJoin(i,j) = 0;
        end
        CohThreshJoin(j,i) = CohThreshJoin(i,j);

        if tJoinRhoPval(i,j) <= 0.001
            RhoThreshJoin(i,j) = tJoinRho(i,j);
        else
            RhoThreshJoin(i,j) = 0;
        end
        RhoThreshJoin(j,i) = RhoThreshJoin(i,j);
    end
end
```

Samples

```
n = 129;

Rho = struct;
Coh = struct;
for i = 1:147
    Rho.(strcat('t',num2str(i))) = zeros(241);
    Rho.(strcat('pVal',num2str(i))) = zeros(241);
    Coh.(strcat('t',num2str(i))) = zeros(241);
    Coh.(strcat('pVal',num2str(i))) = zeros(241);
end

for h = 1:samples
    for i = 1:channels
        for j = i+1:channels

            r = SC.Averages.Rxy.(strcat('Rxy',num2str(h)))(i,j);
            t = r * sqrt( (L1 - 1) / (1 - r^2) );
            Rho.(strcat('t',num2str(h)))(i,j) = t;
                Rho.(strcat('t',num2str(h)))(j,i)...
                    = Rho.(strcat('t',num2str(h)))(i,j);  % Symmetric
            Rho.(strcat('pVal',num2str(h)))(i,j)...
                = 1-tcdf(abs(t),L1-2);
             Rho.(strcat('pVal',num2str(h)))(j,i)...
                 = Rho.(strcat('pVal',num2str(h)))(i,j); % Symmetric

            r = SC.Averages.Coh.(strcat('Coh',num2str(h)))(i,j);
            t = r * sqrt( (n - 1) / (1 - r^2) );

            Coh.(strcat('t',num2str(h)))(i,j) = t;
                Coh.(strcat('t',num2str(h)))(j,i)...
                    = Coh.(strcat('t',num2str(h)))(i,j);  % Symmetric
            Coh.(strcat('pVal',num2str(h)))(i,j)...
                = 1-tcdf(abs(t),n-2);
             Coh.(strcat('pVal',num2str(h)))(j,i)...
                 = Coh.(strcat('pVal',num2str(h)))(i,j); % Symmetric

        end
    end
end
```

Threshold

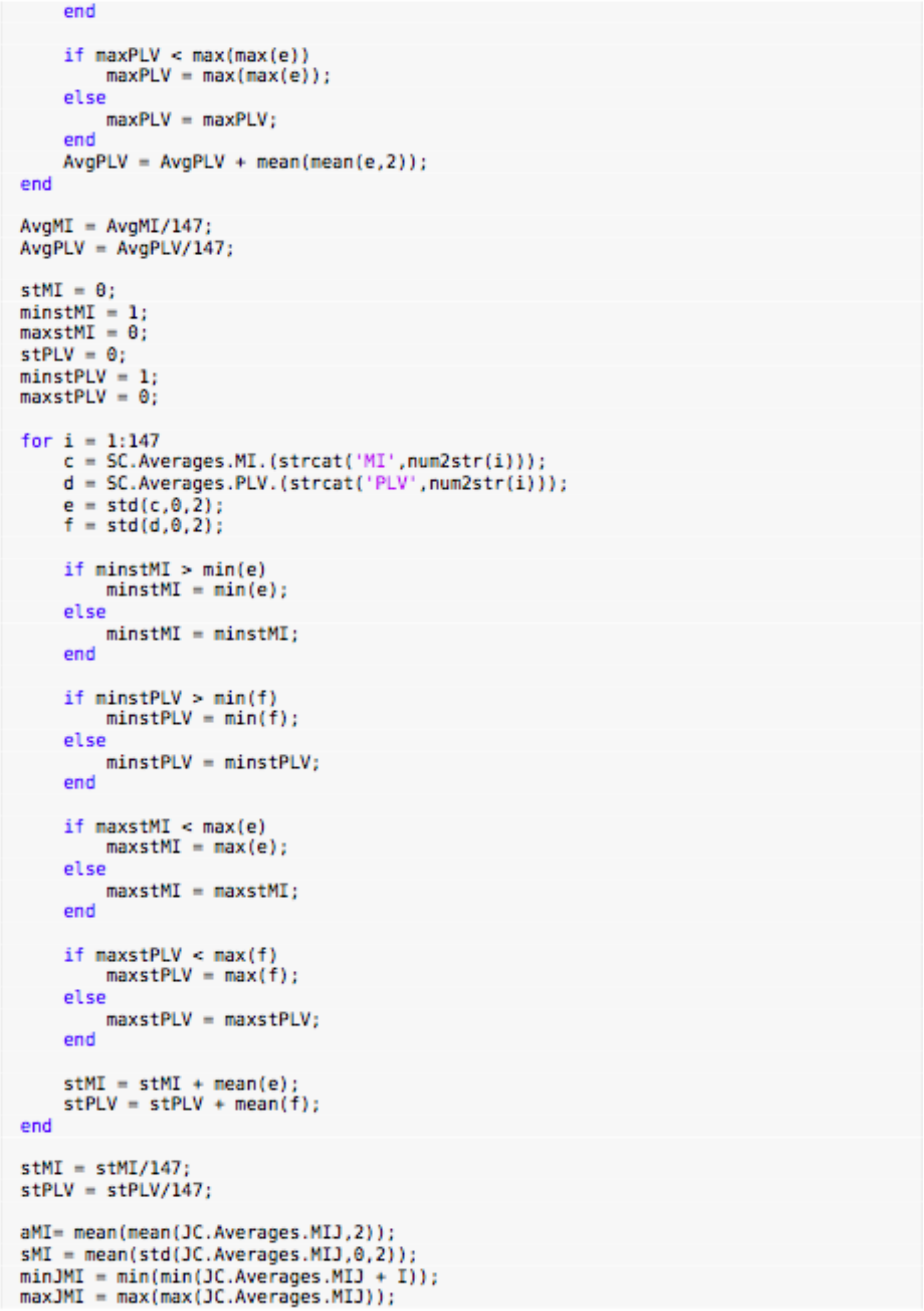
```
RhoThreshSamp = cell(147,1);
CohThreshSamp = cell(147,1);
AvgRhoSamp = zeros(channels);
AvgCohSamp = zeros(channels);
for i = 1:samples
    for ii = 1:channels
        for jj = 1+ii:channels
            if Rho.(strcat('pVal',num2str(i)))(ii,jj) <= 0.05
                RhoThreshSamp{i,1}(ii,jj)...
                    = Rho.(strcat('t',num2str(h)))(ii,jj);
            else
                RhoThreshSamp{i,1}(ii,jj) = 0;
            end
            RhoThreshSamp{i,1}(jj,ii)...
                = RhoThreshSamp{i,1}(ii,jj);

            if Coh.(strcat('pVal',num2str(i)))(ii,jj) <= 0.05
                CohThreshSamp{i,1}(ii,jj)...
                    = Coh.(strcat('t',num2str(h)))(ii,jj);
            else
                CohThreshSamp{i,1}(ii,jj) = 0;
            end
            CohThreshSamp{i,1}(jj,ii) = CohThreshSamp{i,1}(ii,jj);
        end
    end
    AvgRhoSamp = AvgRhoSamp + RhoThreshSamp{i,1};
    AvgCohSamp = AvgCohSamp + CohThreshSamp{i,1};
end
```

```
AvgRhoSamp = AvgRhoSamp./samples;
AvgCohSamp = AvgCohSamp./samples;

% save intermediate results
save('/Users/nacho/Google Drive/TFM/PreResults/Finale/ThresholdedSync.mat',...
    'AvgCohSamp','AvgRhoSamp','avgPLVjoin','avgPLVsamp',...
    'CohThreshJoin','RhoThreshJoin');
```

Look for the significative ts in percentages:

- *X* significative values out of $241^2 \cdot \frac{x}{n} \cdot 100$

```
CohSampSig = zeros(147,1);
RhoSampSig = zeros(147,1);

for i = 1:samples
    CohSampSig(i) = (numel(find(Coh.(strcat('pVal',num2str(i))) <= 0.05)) / Max) * 100;
    RhoSampSig(i) = (numel(find(Rho.(strcat('pVal',num2str(i))) <= 0.05)) / Max) * 100;
end

minMI = 1;
maxMI = 0;
AvgMI = 0;
minPLV = 1;
maxPLV = 0;
AvgPLV = 0;

for i = 1:147

    c = SC.Averages.MI.(strcat('MI',num2str(i)));
    I = eye(241);
    d = c + I;
    if minMI > min(min(d))
        minMI = min(min(d));
    else
        minMI = minMI;
    end

    if maxMI < max(max(c))
        maxMI = max(max(c));
    else
        maxMI = maxMI;
    end

    AvgMI = AvgMI + mean(mean(c,2));

    e = SC.Averages.PLV.(strcat('PLV',num2str(i)));
    f = e + I;

    if minPLV > min(min(f))
        minPLV = min(min(f));
    else
        minPLV = minMI;
    end

    if maxPLV < max(max(e))
```



```
    end

    if maxPLV < max(max(e))
        maxPLV = max(max(e));
    else
        maxPLV = maxPLV;
    end
    AvgPLV = AvgPLV + mean(mean(e,2));
end

AvgMI = AvgMI/147;
AvgPLV = AvgPLV/147;

stMI = 0;
minstMI = 1;
maxstMI = 0;
stPLV = 0;
minstPLV = 1;
maxstPLV = 0;

for i = 1:147
    c = SC.Averages.MI.(strcat('MI',num2str(i)));
    d = SC.Averages.PLV.(strcat('PLV',num2str(i)));
    e = std(c,0,2);
    f = std(d,0,2);

    if minstMI > min(e)
        minstMI = min(e);
    else
        minstMI = minstMI;
    end

    if minstPLV > min(f)
        minstPLV = min(f);
    else
        minstPLV = minstPLV;
    end

    if maxstMI < max(e)
        maxstMI = max(e);
    else
        maxstMI = maxstMI;
    end

    if maxstPLV < max(f)
        maxstPLV = max(f);
    else
        maxstPLV = maxstPLV;
    end

    stMI = stMI + mean(e);
    stPLV = stPLV + mean(f);
end

stMI = stMI/147;
stPLV = stPLV/147;

aMI= mean(mean(JC.Averages.MIJ,2));
sMI = mean(std(JC.Averages.MIJ,0,2));
minJMI = min(min(JC.Averages.MIJ + I));
maxJMI = max(max(JC.Averages.MIJ));
```

```
aCoh = mean(mean(JC.Averages.CohJ,2));
sCoh = mean(std(JC.Averages.CohJ,0,2));
minJCoh = min(min(JC.Averages.CohJ + I));
maxJCoh = max(max(JC.Averages.CohJ));

aPLV = mean(mean(JC.Averages.PLVJ,2));
sPLV = mean(std(JC.Averages.PLVJ,0,2));
minJPLV = min(min(JC.Averages.PLVJ + I));
maxJPLV = max(max(JC.Averages.PLVJ));

aRho = mean(mean(abs(JC.Averages.RxyJ),2));
sRho = mean(std(abs(JC.Averages.RxyJ),0,2));
minJRho = min(min(JC.Averages.RxyJ + I));
maxJRho = max(max(JC.Averages.RxyJ));
```

Change name to get different measures' means

```
meansRho = zeros(1,147);
varsRho = zeros(1,147);
meansPLV = zeros(1,147);
varsPLV = zeros(1,147);
meansMI = zeros(1,147);
varsMI = zeros(1,147);
meansCoh = zeros(1,147);
varsCoh = zeros(1,147);

for i = 1:147
    meansRho(i) = mean(mean(abs(SC.Averages.Rxy.(strcat('Rxy',num2str(i)))),2));
    varsRho(i) = mean(std(abs(SC.Averages.Rxy.(strcat('Rxy',num2str(i)))),0,2));

    meansCoh(i) = mean(mean(SC.Averages.Coh.(strcat('Coh',num2str(i))),2));
    varsCoh(i) = mean(std(SC.Averages.Coh.(strcat('Coh',num2str(i))),0,2));

    meansPLV(i) = mean(mean(SC.Averages.PLV.(strcat('PLV',num2str(i))),2));
    varsPLV(i) = mean(std(SC.Averages.PLV.(strcat('PLV',num2str(i))),0,2));

    meansMI(i) = mean(mean(SC.Averages.MI.(strcat('MI',num2str(i))),2));
    varsMI(i) = mean(std(SC.Averages.MI.(strcat('MI',num2str(i))),0,2));

    meansMI5(i) = mean(mean(SC.Averages.it5_zScMI.(strcat('zScMI',num2str(i))),2));
    varsMI5(i) = mean(std(SC.Averages.it5_zScMI.(strcat('zScMI',num2str(i))),0,2));
end

maxvalCoh = max(meansCoh);
minvalCoh = min(meansCoh);
meanofmeanCoh = mean(meansCoh);

maxvalPLV = max(meansPLV);
minvalPLV = min(meansPLV);
meanofmeanPLV = mean(meansPLV);

maxvalMI = max(meansMI);
minvalMI = min(meansMI);
meanofmeanMI = mean(meansMI);

maxvalMI5 = max(meansMI5);
minvalMI5 = min(meansMI5);
meanofmeanMI5 = mean(meansMI5);
```

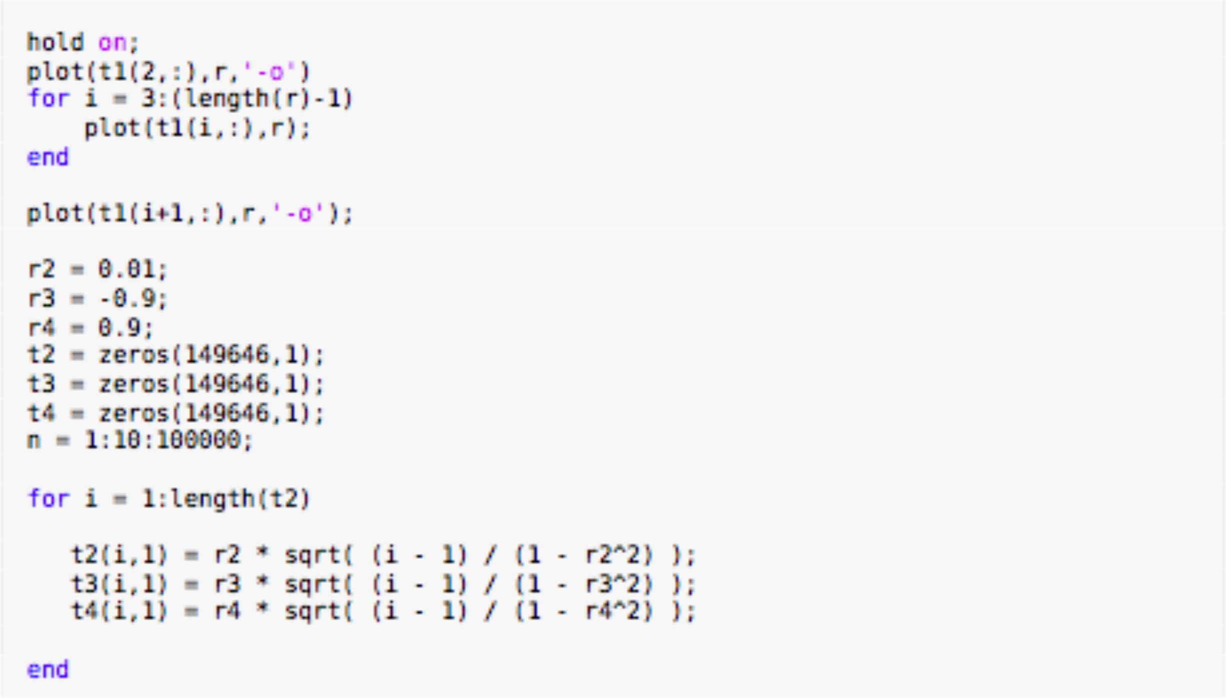
```
maxvalRho = max(meansRho);
minvalRho = min(meansRho);
meanofmeanRho = mean(meansRho);

maxvarCoh = max(varsCoh);
minvarCoh = min(varsCoh);
meanofvarsCoh = mean(varsCoh);

maxvarPLV = max(varsPLV);
minvarPLV = min(varsPLV);
meanofvarPLV = mean(varsPLV);

maxvarMI = max(varsMI);
minvarMI = min(varsMI);
meanofvarsMI = mean(varsMI);

maxvarMI5 = max(varsMI5);
minvarMI5 = min(varsMI5);
meanofvarsMI5 = mean(varsMI5);

maxvarRho = max(varsRho);
minvarRho = min(varsRho);
meanofvarsRho = mean(varsRho);
```

Test for Coherence and Pearson normalization

```
r = -0.9:0.01:0.9;
t1 = zeros(149646,length(r));

for i = 1:149646
    for j = 1:length(r)
        t1(i,j) = r(j) * sqrt( (i - 1) / (1 - r(j)^2) );
    end
end
```

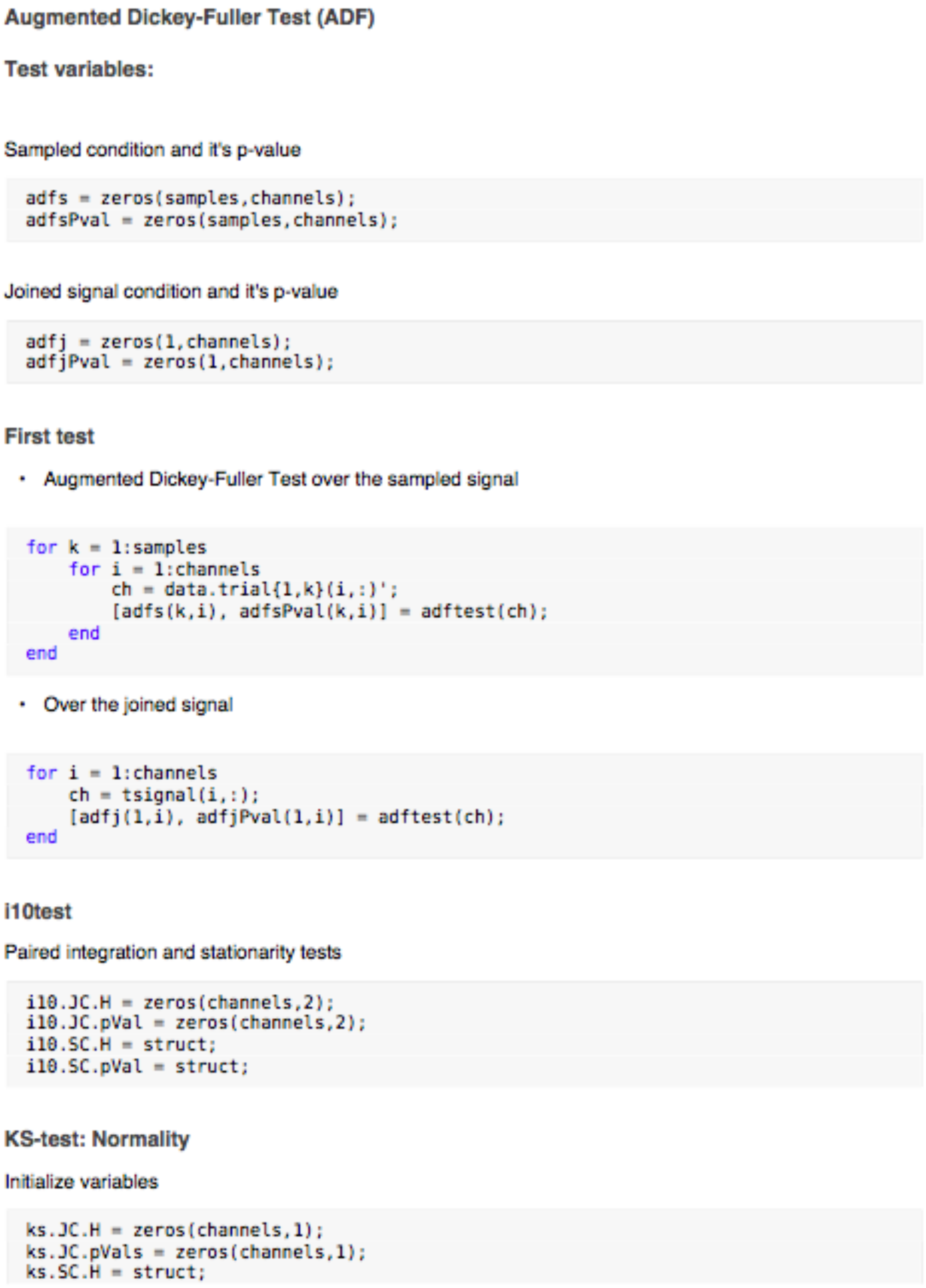
plot results

```
figure
j = 1;
for i = 180:-20:1
    plot(t1(:,i));
    hold on;
    legendInfo{j} = [strcat('r = ', num2str(r(i)))];
    j = j + 1;
end

legend(legendInfo)

figure;
for i = 1:length(t1)
    plot(t1(i,:))
    hold on;
end

figure;
plot(t1(2,:));
```



```
hold on;
plot(t1(2,:),r,'-o')
for i = 3:(length(r)-1)
    plot(t1(i,:),r);
end

plot(t1(i+1,:),r,'-o');

r2 = 0.01;
r3 = -0.9;
r4 = 0.9;
t2 = zeros(149646,1);
t3 = zeros(149646,1);
t4 = zeros(149646,1);
n = 1:10:100000;

for i = 1:length(t2)

   t2(i,1) = r2 * sqrt( (i - 1) / (1 - r2^2) );
   t3(i,1) = r3 * sqrt( (i - 1) / (1 - r3^2) );
   t4(i,1) = r4 * sqrt( (i - 1) / (1 - r4^2) );

end
```

Assumptions Tests



Augmented Dickey-Fuller Test (ADF)

Test variables:

Sampled condition and it's p-value

```
adfs = zeros(samples,channels);
adfsPval = zeros(samples,channels);
```

Joined signal condition and it's p-value

```
adfj = zeros(1,channels);
adfjPval = zeros(1,channels);
```

First test

- Augmented Dickey-Fuller Test over the sampled signal

```
for k = 1:samples
    for i = 1:channels
        ch = data.trial{1,k}(i,:)';
        [adfs(k,i), adfsPval(k,i)] = adftest(ch);
    end
end
```

- Over the joined signal

```
for i = 1:channels
    ch = tsignal(i,:);
    [adfj(1,i), adfjPval(1,i)] = adftest(ch);
end
```

i10test

Paired integration and stationarity tests

```
i10.JC.H = zeros(channels,2);
i10.JC.pVal = zeros(channels,2);
i10.SC.H = struct;
i10.SC.pVal = struct;
```

KS-test: Normality

Initialize variables

```
ks.JC.H = zeros(channels,1);
ks.JC.pVals = zeros(channels,1);
ks.SC.H = struct;
ks.SC.pVal = struct;

for k = 1:samples
    i10.SC.H.(strcat('H', num2str(k))) = zeros(channels,2);
    i10.SC.pVal.(strcat('pVal', num2str(k))) = zeros(channels,2);
    ks.SC.H.(strcat('H', num2str(k))) = zeros(channels,1);
    ks.SC.pVal.(strcat('pVal', num2str(k))) = zeros(channels,1);
end
```

Calculation

```
for k = 1:samples
    for i = 1:channels
       ch = data.trial{1,k}(i,:)';
       [i10.SC.H.(strcat('H', num2str(k)))(i,:),i10.SC.pVal.(...
           strcat('pVal', num2str(k)))(i,:)] ...
           = i10test(ch,'display','off');
       [ks.SC.H.(strcat('H', num2str(k)))(i),ks.SC.pVal.(...
           strcat('pVal', num2str(k)))(i)] = kstest(ch);
    end
end

for i = 1:channels
    ch = tsignal(:,i);
    [i10.JC.H(i,:),i10.JC.pVal(i,:)] ...
        = i10test(ch,'display','off');
    [ks.JC.H(i),ks.JC.pVal(i)] = kstest(ch);
end
```

NonLinearity Tests: Boottime (functions obtained from:

https://es.mathworks.com/matlabcentral/fileexchange/16062-test-of-non-linearity)

posted by the author (Barnett)

Initialize variables

```
NlT.JC = struct;
NlT.SC = struct;

NlT.SC.s1 = struct;
NlT.SC.s.Region = struct;
NlT.SC.s.JackStats = struct;

for l = 1:samples
    for i = 1:channels
        NlT.SC.(strcat('s',num2str(l))).(strcat('Region',num2str(i))) = [];
        NlT.SC.(strcat('s',num2str(l))).(strcat('JackStats',num2str(i))) = [];
    end
end

for i = 1:channels
   NlT.JC.(strcat('Region',num2str(i))) = [];
   NlT.JC.(strcat('JackStats',num2str(i))) = [];
end
```

Calculation

```
for k = 1:samples
    k
    ch = tsignal(:,i);
    for i = 1:channels
        [
            NlT.SC.(strcat('s', num2str(k))).(...
            strcat('Region',num2str(i))),...
            NlT.SC.(strcat('s',num2str(k))).(...
            strcat('JackStats',num2str(i)))...
        ] = boottime(ch,3,500,0.05,1);
    end
end
```

TEST RESULTS (average results)

First Augmented Dickey-Fuller Test

```
% Logic vector (1 not rejected, 0 rejected) Rejections (in each sample)
rsA.H = struct;
% Number of not-rejections (in each sample)
nrA = zeros(147,1);
% Logic vector (1 p > 0.01, 0 = p<0.01)
psA.pVal = struct;
% Number of significative pVals (in each sample)
npA = zeros(147,1);
```

Kwiatkowski PSS test

```
% Logic vector (1 not rejected, 0 rejected) Rejections (in each sample)
rsK.H = struct;
% Number of not-rejections (in each sample)
nrK = zeros(147,1);
% Logic vector (1 p > 0.01, 0 = p<0.01)
psK.pVal = struct;
% Number of significative pVals (in each sample)
npK = zeros(147,1);
% Comparison to know if the rejected cases correspond to the p vals < 0.01
eqA = zeros(147,1);
eqK = zeros(147,1);
```

Find signficant results

```
for i = 1:samples
    rsA.(strcat('H',num2str(i))) = i10.SC.H.(strcat('H',num2str(i)))(:,1)~=0;
    nrA(i) = numel(find(rsA.(strcat('H',num2str(i)))));
    psA.(strcat('pVal',num2str(i))) = i10.SC.pVal.(strcat('pVal',num2str(i)))(:,1)<0.01;
    npA(i) = numel(find(psA.(strcat('pVal',num2str(i)))));
    eqA(i,1) = isequal(rsA.(strcat('H',num2str(i))),psA.(strcat('pVal',num2str(i))));
    rsK.(strcat('H',num2str(i))) = i10.SC.H.(strcat('H',num2str(i)))(:,2)~=0;
    nrK(i) = numel(find(rsK.(strcat('H',num2str(i)))));
    psK.(strcat('pVal',num2str(i)))...
        = i10.SC.pVal.(strcat('pVal',num2str(i)))(:,2)<0.05;
    npK(i) = numel(find(psK.(strcat('pVal',num2str(i)))));
    eqK(i,1) = isequal(rsK.(strcat('H',num2str(i))),psK.(strcat('pVal',num2str(i))));
end
```

Check results of nonlinearity:

Initialize

```
% Number of not-rejections (in each sample)
nrL = zeros(147,1);
% Number of significative pVals (in each sample)
npL = zeros(147,1);
% Comparison to know if the rejected cases correspond to the p vals < 0.01
eqL = zeros(147,1);

for i = 1:147
    for k = 1:channels
        ResSC.(strcat('s', num2str(i)))(k,:)...
            = NLT.SC.(strcat('s',...
            num2str(i))).(strcat('JackStats', num2str(k)))(1,4:5);
    end
end

numRejSC = zeros(147,1);
numPvalSC = zeros(147,1);
```

Look for the meat!

```
for i = 1:147
    idRej.(strcat('s', num2str(i)))...
        = find(ResSC.(strcat('s', num2str(i)))(:,2)~=0);
    idPval.(strcat('s', num2str(i)))...
        = find(ResSC.(strcat('s', num2str(i)))(:,1)<0.05);

    numRejSC(i) = numel(idRej.(strcat('s', num2str(i))));
    numPvalSC(i) = numel(idPval.(strcat('s', num2str(i))));
end
plot(numPvalSC,'o') %% Plot it
ResJC = zeros(channels,2);
for i = 1:241
    ResJC(i,:) = NLT.JC.(strcat('JackStats', num2str(i)))(1,4:5);
end

idRejJC = find(ResJC(:,2)~=0);
idPvalJC = find(ResJC(:,1)>0.05);
numRejJC = numel(idRejJC);
numPvalJC = numel(idPvalJC);
% Number of significantly rejected
GoodRejJC = find(ResJC(:,2) ~= 0 & ResJC(:,1) < 0.05);
GoodnumPvalJC = numel(GoodRejJC);
```

Signal to Noise Ratio

```
snrJoin = mean(tsignal,2)./std(tsignal,0,2);
```

```
snrSamp = zeros(channels,samples);
for i = 1:samples
    snrSamp(:,i) = mean(data.trial{1,i},2)./std(data.trial{1,i},0,2);
end
SNR = struct;
SNR.Join = snrJoin';
SNR.Samples = snrSamp;
```

Describing SNR (channels)

```
[msJ,a] = min(snrJoin');
[MsJ,b] = max(snrJoin');
AvsJ = mean(snrJoin);
StsJ = std(snrJoin);


AvsS = mean(mean(snrSamp,2));
StsS = mean(std(snrSamp,0,2));
[msS,c] = min(min(snrSamp'));
[MsS,d] = max(max(snrSamp'));
```

```
plot(snrJoin,mean(snrSamp,2),'o')
```