Development of a partial correlation algorithm to estimate functional connectivity in cortical networks

Academic year 2013/2014

SUPERVISOR
Prof. Paolo Massobrio

ASSISTANT SUPERVISOR
Dott.Ing. Daniele Poli

APPLICANT
Vito Paolo Pastore
Se poni una piccola cosa su un'altra piccola cosa, e fai questo continuamente, presto avrai una cosa grande.

(Esiodo, Opere e giorni, 361 s.)
## Index

ABSTRACT ................................................................................................................................. 5  
1. INTRODUCTION ................................................................................................................... 7  
2. MATERIALS AND METHODS .............................................................................................. 11  
   2.1 CROSS-CORRELATION .................................................................................................. 11  
      2.1.1 A SPECIAL CASE OF CROSS-CORRELATION: THE PSTH .................................. 13  
      2.1.2 CONDITIONAL FIRING PROBABILITY .................................................................... 14  
      2.1.3 ACTIVITY PAIRS ...................................................................................................... 16  
      2.1.4 OUR APPROACH TO COMPUTE THE CROSS-CORRELATION ............................ 17  
      2.1.5 OTHER POSSIBLE APPROACHES .......................................................................... 18  
      2.1.6 EXTRACTION OF USEFUL FUNCTIONS AND PARAMETERS .............................. 19  
   2.2 PARTIAL CORRELATION .................................................................................................. 25  
   2.3 TRANSFER ENTROPY ...................................................................................................... 28  
   2.4 CELL CULTURE, EXPERIMENTAL SET-UP AND DATA ANALYSIS ............................. 29  
   2.5 MICRO-ELECTRODES-ARRAYS (MEA) ....................................................................... 30  
   2.6 NETWORK MODEL DESCRIPTION ............................................................................... 32  
   2.7 VALIDATION PROCEDURE ............................................................................................ 33  
      2.7.1 ROC CURVE ............................................................................................................... 34  
      2.7.2 THRESHOLD SELECTION ....................................................................................... 36  
   2.8 CONNECTIVITY MAP ....................................................................................................... 40  
   2.9 DESCRIPTION OF THE IMPLEMENTED ALGORITHM ..................................................... 42  
      2.9.1 DATA ADAPTATION AND PROCESSING ............................................................... 42  
      2.9.2 COMPUTING THE SPECTRUM OF CROSS-CORRELATION .................................. 43  
      2.9.3 COMPUTING THE SCALED PARTIAL COVARIANCE DENSITY (SPCD) ............... 44  
      2.9.4 OUTPUT PROCESSING .............................................................................................. 44  
      2.9.5 GRAPHIC INTERFACE ............................................................................................. 45  
      2.9.6 SCHEMATIC DESCRIPTION OF THE IMPLEMENTED ALGORITHM ...................... 47  
      2.9.7 APPENDIX: MATHEMATICAL TOOLS USED IN THE ALGORITHM .................... 48  
      2.9.8 COMPUTATIONAL LIMITS OF THE ALGORITHM .................................................. 49  
3. RESULTS AND DISCUSSION ................................................................................................. 51  
   3.1 THRESHOLD SELECTION ................................................................................................. 51  
      3.1.1 HEURISTIC THRESHOLD VERSUS MCC/ACC THRESHOLD .................................. 55  
   3.2 ROC CURVES .................................................................................................................. 57
3.3 ACC CURVE ........................................................................................................................................59
3.4 MCC CURVE ........................................................................................................................................62
4. FUTURE PERSPECTIVES .........................................................................................................................64
  4.1 APPLICATION OF THE IMPLEMENTED ALGORITHM TO EXPERIMENTAL DATA .................64
  4.2 EXTENSION OF THE ALGORITHM TO A LARGER NUMBER OF NEURONS ..................65
5. REFERENCES ...........................................................................................................................................67
ABSTRACT

The main goal of this work was to develop an algorithm able to perform a cross and partial correlation analysis in order to identify functional connectivity in cortical networks. A detailed analysis of functional connectivity of *in vitro* neural networks, as well as the possibility to understand the interplay between topology, structure, function and dynamics, is very important for better understanding how the central nervous system represents and stores the information (i.e. the *neural code*). After the implementation of the algorithm, we tested it on simulated networks by varying the average degree of connectivity from 10 (neurons low connected) to 60 (neurons fully connected), and considering 20 simulations per degree of connectivity. In order to evaluate the algorithm’s performance, we used a specific validation procedure based on three different metrics: Receiver Operating Characteristic (ROC) curve, the Matthews Correlation Coefficient (MCC) and the Accuracy coefficient (ACC). Finally we applied the implemented algorithm to an *in vitro* cortical network, building the connectivity graph of the analysed network.
ABSTRACT (VERSIONE ITALIANA)

Lo scopo principale della tesi è stato sviluppare un algoritmo in C# che implementasse un’analisi basata su cross-correlazione e correlazione parziale, per ricostruire la connettività funzionale in reti di neuroni. Infatti l’analisi dettagliata della connettività funzionali delle reti neuronali in vitro, così come la possibilità di comprendere la relazione che sussiste tra struttura, funzione e dinamica, è fondamentale per capire in che modo il sistema nervoso centrale rappresenta e memorizza le informazioni (il codice neurale). Dopo l’implementazione, abbiamo testato l’algoritmo su reti simulate, variando il grado medio di connettività da 10 (neuroni poco connessi) a 60 (neuroni completamente connessi); in particolare abbiamo considerato 20 simulazioni per grado di connettività. Per valutare le performance dell’algoritmo, abbiamo utilizzato una procedura di validazione basata su 3 diverse metriche di giudizio: La curva ROC (Receiver Operating Characteristic), il coefficiente di correlazione di Matthews (MCC) e il coefficiente di Accuratezza (ACC). Infine abbiamo applicato l’algoritmo implementato ad una rete corticale in vitro, ricostruendone il grafo di connettività.
1. INTRODUCTION

Since the nineteenth century, we know that the neuronal elements of the brain constitute a
formidably complex structural network (Bullmore et al, 2009). While a large number of
anatomical studies of the human brain have been carried out at the macroscopic (cerebral
lobes, surface landmarks, and white matter tracts) or microscopic (cytoarchitectonics,
myeloarchitectonics, chemoarchitectonics, etc.) level, there is little information about the
organization and function of neural networks (Sporns et al, 2005).

One of the central concepts of contemporary neuroscience is that major brain functions are
executed through the joint actions of neurons. In addition, individual elements of information
are encoded not by single cells, but rather by a population of cells and/or dynamically
instantiated cell assemblies (Nicolelis et al, 1997).

Indeed, one of the major challenge of contemporary neuroscience is approaching the
complex interplay between single cells, small cell networks (i.e. microcircuits) and large
population cell assemblies (Eichler et al, 2003).

Moreover, large-scale cortical networks in the mammalian brain, comprising anatomically
distinct regions and inter-regional pathways, exhibit specific non-random connection
patterns. The structural features of these networks are clearly linked to aspects of brain
function and dynamics, playing a critical role in determining which functional patterns (and
thus, brain states) can and cannot occur (Salinas and Sejnowski, 2001). Hence, it is of
fundamental importance to understand the functioning of neuronal networks by
determining, knowing and characterizing their elements and interconnections (i.e. the human
"connectome") (Eichler et al, 2003).

Since the system under study (i.e. brain areas, cell assemblies) is highly complex, to achieve
the purpose described above, it is useful to adopt a reductionist approach.

A possible strategy to reduce such a complexity makes use of in vitro experimental models
with different architectures (e.g. degree of complexity) coupled to Micro-Electrode Arrays
(MEAs). Large random networks of cortical neurons developing in vitro and chronically
coupled to MEAs (Taketani and Baudry, 2006) represent a well-established experimental
model for studying the neuronal dynamics at the network level (Eytan and Marom, 2006),
and for understanding the basic principles of information coding (Ricke et al, 1997), learning and memory (Marom and Shahaf, 2002). In addition, these model systems represent a good tradeoff between controllability/observability and similarity to the in vivo nervous system and allow accessibility and extensive manipulation both from the chemical and the electrical point of view (Raichman et al, 2008).

The use of MEA allows simultaneous recordings from hundreds of microelectrodes, giving the opportunity to access several "nodes" of the neural networks. From these recordings, it is possible to study how neurons are connected each other, and which topological architectures underlie a specific dynamic behaviour (Teramae et al, 2007). Within this topic, recent technological efforts (i.e. increasing the number of electrodes and the spatial resolution), allow to obtain a more precise mapping of the network, up to a possible identification of its anatomical connections (i.e. the set of physical or structural-synaptic connections linking neuronal units at a given time). In particular, the active pixel sensor array (Berdondini et al, 2009) with 4096 electrodes and the high-density CMOS array (Matsuda et al, 2013) with 11,000 electrodes, are the latest and the most relevant in vitro micro-electrodes recording’s technologies.

As previously stated, the anatomical and physiological observations and analysis of large-scale neuronal networks, give important information to address the problem of linking networks' structure to ongoing and evoked brain dynamics. Moreover, these networks provide a comprehensive description of the brain’s structural connectivity, also called the human connectome (Sporns, 2012). The connectome essentially comprises a complete map of the brain’s structural connections. These structural connections shape large-scale neuronal dynamics which can be captured as patterns of functional connectivity; thus it becomes fundamental to investigate how neurons are functionally connected (Friston et al 2011, Smith et al 2012). Within this general framework, several approaches can be followed, depending on the scale at which the nervous system is observed (ranging from the whole brain to the single synapse level and being neuronal network a good trade-off between complexity and organization). Functional imaging or optical methods, such as fluorescent techniques, could be a possible strategy to achieve the previously described goal, for in vitro preparations.
However, there are some drawbacks related to the limited access to single units and large populations at the same time, and to a poor temporal resolution (Grinvald et al, 2004).

A different approach relies on the identification of causal relationships between pairs of neurons by means of electrophysiological measurements: this complementary method plays a relevant role in the study of synaptic interactions at microcircuit and at population level. Nowadays, a promising technique to infer connectivity maps of a cell culture seems to rely on an investigation of the statistical properties of the spontaneous activity. This technique, also called functional connectivity method, relies on the pair-wise spiking activities of the neurons (Garofalo et al, 2009). Functional connectivity is generally derived from time series observations, and describes patterns of statistical dependence among neural elements (Sporns et al, 2013); the presence of a statistical relationship between two neural elements is considered as a sign of functional coupling. Functional connectivity can be computed in several ways, including cross and partial correlation, mutual information, transfer entropy or spectral coherence. Cross correlation measures the frequency at which one cell fires as a function of time relative to the firing of a spike in another cell (Salinas and Sejnowski, 2001). Mutual information represents a measure of the statistical dependence between two spike trains recorded by two microelectrodes (Xu et al, 1997). Partial correlation analysis allows one to distinguish between direct and indirect connectivities by removing the portion of the relationship between two neural spike trains that can be attributed to linear relationships with recorded spike trains from other neurons (Eichler et al, 2003). Spectral coherence measures the linear time-invariant (LTI) relationship between two time series (e.g. simultaneous recorded spike trains) (Sun et al, 2004).

Transfer Entropy is an information theoretic measure able to estimate causal relationships from time series taking into account their past activity. In particular, TE estimates the part of a neuron activity which does not depend on its own past, but which depends on another neuron’s past activity (Lungarella et al, 2007).

In order to understand the complex relationship between neural dynamics and neuronal network's structure, another fundamental topic of analysis regards the topological architectures which underlie a specific dynamic behaviour (Friston 1994). In fact the collective
dynamics of a network of neurons (and more general of coupled excitable systems) depends on connections’ topology in the network. In a review, Albert and Barabasi (Albert and Barabasi 2002) show that complex systems belonging to different fields, ranging from computer science to biological world, such as neuronal networks, can be modeled by three main stereotyped network architectures called: scale-free (Barabasi and Bonabeau 2003), where most nodes have few connections between each other and only few nodes called hubs are heavily connected with the other nodes/cells; random (Barabasi and Albert 1999), where the links connectivity between cells/nodes follow a poisson distribution; small-world (Watts and Strogatz 1998), which describes totally regular networks rewired in order to introduce a small degree of disorder.

The description of neural networks by means of the aforementioned network's topologies, implies to consider the brain as a complex network of interacting dynamical systems (Varela et al, 2001). Thus, it allows to describe structural brain networks using the graph theory. Graph theory is a branch of mathematics that deals with the formal description and analysis of graphs (Bullmore et al, 2009). Within the framework of this theory, structural and functional brain networks can be described as graphs composed of nodes (vertices), denoting neural elements (neurons or brain regions), linked by edges representing physical connections (synapses or axonal projections).

Generally speaking, neural networks in the brain present locally clustered connections with a small admixture of long-range connections. The coexistence of high degrees of local clustering and short path length (i.e. the extension of the physical connection, the edges' length in the network's graph) is peculiar of small-world network topologies (Downs et al, 2012), nowadays considered as the best candidate to support the emergence of two fundamental organizational principles in the nervous system: functional segregation and functional integration. Segregation (or specialization) refers to the degree to which a network’s elements form separate cliques or clusters. Integration refers to the capacity of the network as a whole to become interconnected and exchange information (Sporns et al, 2013). Segregation and Integration are respectively associated to local clusters and long-range connections found in neural networks (Zalesky et al, 2012).
2. MATERIALS AND METHODS

2.1 CROSS-CORRELATION

In neuroscience the cross-correlogram function is constructed from the spike trains of two neurons, and it measures the frequency at which one cell fires (i.e. the ‘target’) as a function of the time relative to the firing of a spike in another cell (i.e. the ‘reference’) (Salinas and Sejnowski, 2001).

Mathematically (Knox, 1981), the correlation function is a statistic representation of the average value of the product of two stochastic processes which, in our case, are spike trains. Let us consider a neuronal population $V$ and two specific neurons, $a, b \in V$. One simple indicator of functional connectivity between $a$ and $b$ is the standard correlation $R_{ab}(\tau)$ of their associated spike trains: $x_a(t)$ and $x_b(t)$. Here, the Cross-correlation function reduces to a simple probability $R_{ab}(\tau)$ of observing a spike in one train $x_a(t)$ at time $t + \tau$, given that there is a spike in a second train $x_a(t)$ at time $t$ (Rieke et al, 1997). $\tau$ is called the time shift or the time lag. To generate a cross-correlogram between the two spike trains $x_a(t)$ – chosen as the reference train and $x_b(t)$ – chosen as the target train, the two trains are lined up and a single spike in the $x_a$ train selected to mark the point $t=0$ (see Figure 1). The latencies of all the $x_b$ spikes to the right or left of this, in a time window of hundreds of milliseconds (e.g. 150 ms), are then computed and plotted as a histogram. This process is repeated using the next spike in the $x_a$ train as the origin (i.e. the point $t=0$) and so on. When all the $x_a$ spikes have been considered, the accumulated counts in each bin will be proportional to the average probability of observing $x_b$ spike before or after any arbitrary $x_a$ spike. In other words, the cross-correlogram shows a count of the spikes of the target cell at specific time delays with respect the spikes of the reference cell. The time delay is given by distance along the horizontal axis, and both ‘positive’ and ‘negative’ delays are recorded. Bin counts on the positive side (i.e. counts occurring after time 0, the windows center) mean that the target cell spike came after the reference cell spike, conversely for the negative side. When two spike trains are independent, the cross-correlogram is flat; if there is any covariation in the spike trains, one or more peaks appear (Brody, 1999).
Figure 1: Scheme depicting how the cross-correlogram is generated. The origin of the cross-correlogram is aligned with a spike in the reference train. Counts of one are then added to the correlogram at times of occurrence of the target spikes. The correlogram is shifted to the next reference spike and the process repeated. When all references spikes have been used, the cross-correlogram gives the average probability of observing a spike in the target train, before or after any arbitrary spike in the reference train.

Figure 2: The cross-correlogram shows a count of the spikes of the target cell at specific time delays $\tau$ with respect the spikes of the reference cell; the time delay is given by the distance along the horizontal axis, and both ‘positive’ and ‘negative’ delays are recorded.
2.1.1 A SPECIAL CASE OF CROSS-CORRELATION: THE PSTH

In many neurophysiological experiments a controlled series of ‘changes’ in the physical environment can be introduced, such as repeated, relatively short, trains of stimuli (Perkel et al, 1967). To detect and evaluate the effect of such stimulation on the spike trains, it is common and very useful to compute a Post-Stimulus-Time Histogram (PSTH) (Gerstein and Kiang, 1960).

Spike trains in response to a repeated presentation of the same stimulus are not identical, so the PSTH represents the **average behaviour of one or more cells to a stimulation pattern**. More specifically, the PSTH shows the probability of firing as function of time after the stimulus onset. If the stimulus has no effect on the pattern of the spike train, the PSTH will be flat (subject to the usual statistical fluctuations). On the other hand, if the stimulus does produce a time locked “evoked response” in the spike train pattern, the PSTH will show deviations from flatness. A peak in the PSTH indicates a higher probability of firing at that particular time after stimulation and can presumably be associated with an excitatory process. Dips in a PSTH indicate a lower time locked probability of firing and often are associated with inhibitory or refractory processes.

The mathematical definition of the PSTH is reported below:

\[
P(\tau) = \frac{1}{N} \sum_{i=1}^{N} x(\tau - t_{STi})
\]

*Eq. 1*

where \( N \) is the total number of stimuli delivered to the network, \( t_{STi} \) is the timing at which each stimulus \( i \) occurs and \( x \) is the spike train recorded at a specific channel. The time-axis is generally divided in bins of amplitude \( \Delta \tau \). If the histogram is normalized by the number of stimulus presentation and also by the bin size (not reported in the above formula), the resulting PSTH gives the firing rate - or the probability per unit time of firing, if each bin contains at most 1 spike- as a function of time.
Figure 2: Variability of neural response and construction of the average response an example of PSTH is reported (Rieke et al., 1997). The top panel shows a raster plot of 50 individual spike trains in response to a stimulus at $t = 0$. Each dot in the raster plot marks the time of occurrence of a single spike. In this case, spikes are recorded extracellularly from the movement sensitive H1 in the fly visual system. The lower panel represents the relative PSTH, as described above, which, in the most general case, gives the firing rate as a function of time. The delay before the peak in the firing rate depicted in the PSTH is due to the delays in the visual receptors and in the synapses between the receptors and H1.

2.1.2 CONDITIONAL FIRING PROBABILITY

The conditional firing probability $CFP_{ij}(\tau)$ is defined by the incidence of an action potential at electrode $j$ at delay $\tau$ ($0 \leq \tau \leq 500$) after that a spike has been detected at electrode $i$, divided by the total number of action potentials at $i$ (le Feber et al., 2007). In the approach devised by Le Feber, it is very innovative the use of parameters extracted by fitting the obtained function with a particular curve, as we will shortly see. To analyse the measured signals, binary arrays $X_i$ were constructed for all recording sites $i$, with as many data points as the sampled signals, with $X_i[n] = 1$ at a detected action potential and $X_i[n] = 0$ elsewhere. The number of
action potentials at electrode \( i \) that is followed by a spike at \( j \) with a delay \( \tau \) \((N_{\text{follow},ij}[\tau])\) is now calculated as:

\[
N_{\text{follow},ij}[\tau] = \sum_t X_i[t] \cdot X_j[t + \tau]
\]

\( Eq. 2 \)

Equation 2 holds because it is applied to binary arrays \( X_i \) and \( X_j \) with \( X_{ij}[n] \in [0, 1] \) for all \( n \).

\( CFP \) \((\tau)\) can be calculated by dividing \( N_{\text{follow},ij}[\tau] \) by the total number of action potentials at electrode \( i \) \((N_i)\):

\[
\text{CFP}_{ij}[\tau] = \frac{N_{\text{follow},ij}[\tau]}{N_i} = \frac{\sum_t X_i[t] \cdot X_j[t + \tau]}{\sum_t X_i[t]}
\]

\( Eq. 3 \)

It may be noted that \( \text{CFP}_{ij}[\tau] \) is a measure related to cross-correlation \((R_{ij}[\tau])\):

\[
R_{ij}[\tau] = \frac{1}{L} \sum_{i=1}^{N} X_i[t] \cdot X_j[t + \tau]
\]

\( Eq. 4 \)

where \( L \) is the length (the number of elements) of the interval in which \( X_i \) and \( X_j \) are compared.

Thus,

\[
\text{CFP}_{ij}[\tau] = \frac{R_{ij}[\tau]}{\mu_i}
\]

\( Eq. 5 \)

Where \( \mu_i \) \((=1/L \sum X_i)\) equals the average of \( X_i \) which is proportional to the number of action potentials recorded at electrode \( i \) in the interval of length \( L \).

If \( \text{CFP}_{ij}[\tau] \) showed a distribution that clearly deviated from flat, electrodes \( i \) and \( j \) were considered related.
Figure 3: Example of Conditional Firing probability curve between two electrodes correlated. In the inset there is an example of flat distribution, meaning that the two analysed electrodes are not correlated.

2.1.3 ACTIVITY PAIRS

The underlying assumption of Eytan and Marom’s approach (Eytan et al, 2004) is that associations between neuronal activities, or some trivial extension of such, are the neurophysiological building blocks of behavioural associations. Operationally, they define pairs of diachronically associated spikes, denoted for simplicity as activity pairs, in terms of an action potential A that is followed by another action potential B with a precise time delay (τ ± Δτ) milliseconds between the two. Thus, for instance, for 0< τ<150 msec, Δτ = 2.5msec, there are total of 30 activity pairs for a given A -> B. A and B may be action potentials recorded from the same or different electrodes; in each network they identify thousands of such pairs. In some instances, they use an external stimulation as the first element of a pair in which case they denote the pair S->R (instead of A -> B). For each activity pair, with a time delay τ±Δτ msec between the elements of the pair, they define functional association strength \( C(\tau) \) as the number of occurrences in which action potential A was followed by action potential B within τ±Δτ, divided by the number of occurrences of A measured within a time period. The functional association strength is related to a correlation measure (with a time lag of τ) between two neuronal activities. This
measure can be intuitively interpreted as the efficacy of $A$ in predicting the activity of $B$. Thus defined, each diachronically associated pair of spikes represents a different subset of pathways in the network (Abeles, 1991): pairs become representative of activation paths.

2.1.4 OUR APPROACH TO COMPUTE THE CROSS-CORRELATION

The method we used for computing the cross-correlation function $C_{xy}(\tau)$ is a mixture between the classical definition of cross-correlation between spike train (Knox, 1981; Rieke et al, 1997) and the approach proposed by Eytan and Marom (Eytan et al., 2004). Given two spike trains (i.e. $x$ and $y$), recorded from two electrodes of the MEA, we counted the number of spikes in the $y$ train within a time frame around the spikes of the $x$ train of $\pm T$ ($T$ usually set at 150msec), using bins of amplitude $\Delta \tau$ (usually set at multiple of the sampling frequency).

The correct $C_{xy}(\tau)$ is obtained by means of a normalization procedure, by dividing each element of the array by the square root of the product between the number of peaks in the $x$ and the $y$ train, according to the following formula:

$$C_{xy}(\tau) = \frac{1}{\sqrt{N_x N_y}} \sum_{s=1}^{N_s} \sum_{t=1}^{N_x} x(t_s) y(t_s-t_i)$$

Eq. 6

where $t_s$ indicates the timing of a spike in the $x$ train, $N_x$ is the total number of spikes in the $x$ train and $N_y$ is the total number of spikes in the $y$ train. We made this choice so that the $C_{xy}(\tau)$ belongs to the interval $[0, 1]$, where the value ‘1’ is generally reached only in the case of an autocorrelation and it corresponds to the value of the peak at zero lag. Moreover, the symmetry between $C_{xy}(\tau)$ and $C_{yx}(\tau)$ is maintained since: $C_{xy}(\tau) = C_{yx}(-\tau)$. In this way, many of the parameters that we want to extract from the cross-correlogram are symmetric and the computation can be faster (only half of the possible pairs of electrodes needs to be calculated).
2.1.5 OTHER POSSIBLE APPROACHES

There are other possibilities for normalizing the cross-correlogram:

1. Normalization by the **number of spikes of the x train**:

\[
C_{xy}(\tau) = \frac{1}{N_x} \sum_{s=1}^{N_x} \sum_{t_i=(\tau-\Delta \tau/2)}^{(\tau+\Delta \tau/2)} x(t_s)y(t_s-t_i)
\]

\[Eq. 7\]

This is the normalization method chosen by Eytan and Marom and it is also proposed as the conventional normalization method (Knox, 1981). Thus defined, the correlation measure is
physiologically interpretable as the strength of entailment of \( y \) by \( x \) (i.e. effect that the reference spike train \( x \) has on the spike train \( y \)). Since there is a stochastic element in the neuronal activity, a normalization method of this kind makes \( C(\tau) \) sensitive in cases of small number of trials (occurrences of \( x \) spikes in \( C(\tau) \)). In order to circumvent this problem, when comparing \( C(\tau) \)'s between different recording phases, the following criteria must be included: (i) a threshold on the minimum allowed mean firing rate for the train \( x \) and (ii) the activity couple \( xy \) must appear more than \( n \) times during each of two compared recording phases.

2. Normalization by the **number of spikes of the \( y \) train**: 

\[
C_{xy}(\tau) = \frac{1}{N_y} \sum_{s=1}^{N_y} \sum_{t_i = (\tau - \Delta \tau)}^{(\tau + \Delta \tau)} x(t_s)y(t_s - t_i) 
\]

Eq. 8

This normalization method takes into account only the activity of \( y \), ignoring the activity of \( x \). It has no physiological meaning and it is usually avoided.

3. Normalization by the **product of spikes of the \( x \) and \( y \) train**: 

\[
C_{xy}(\tau) = \frac{1}{N_xN_y} \sum_{s=1}^{N_x} \sum_{t_i = (\tau - \Delta \tau)}^{(\tau + \Delta \tau)} x(t_s)y(t_s - t_i) 
\]

Eq. 9

This normalization method takes into account the activity of both \( x \) and \( y \), but the maximum value of the \( C_{xy}(\tau) \) is not 1 also in case of an autocorrelation and we are not able to interpret the obtained results.

2.1.6 EXTRACTION OF USEFUL FUNCTIONS AND PARAMETERS

Once computed, the cross-correlogram can be useful for extracting parameters aimed at characterizing the behaviour, the synchrony and to build connectivity maps of the network under experiment. In the following, I will present some of them:
Mean cross-correlogram

To get qualitative information on how the activity of an electrode $x$ is correlated to the activity of all the other electrodes $y$ (with $y \neq x$), we can define the mean correlogram, according to the definition below:

$$C_x(\tau) = \frac{1}{n-1} \sum_{y=1}^{n} C_{xy}(\tau)$$

\text{Eq. 10}

for $x \neq y$ and $1 \leq x \leq n$, where $n$ is the number of electrodes analysed.

This function is particularly useful when the global behaviour of the network in different experimental conditions has to be explored: it can be plotted in order to see how each channel is globally correlated to all the others.

$C(0)$

The $C(0)$ (see figure 3) represents the value of the cross-correlogram in an area around the zero bin (i.e. the central bin of the cross-correlation function). This parameter is evaluated in order to quantify the synchronization level among all the recordings channel

$$C(0) = \sum_{\tau = -k*\Delta \tau/2}^{k*\Delta \tau/2} C_{xy}(\tau)$$

\text{Eq. 11}

Where $\Delta \tau$ is the bin size of the correlogram function and $k$ is the number of bins around zero.

$C_{\text{peak}}$

The $C_{\text{peak}}$ represents the value of the cross-correlogram in an area around the maximum detected peak (see figure 3). This coefficient is evaluated in order to quantify the correlation level among all the recording channels (e.g. two channels can be highly correlated with a time interval of a few msec, meaning that their $C(0)$ is low but their $C_{\text{peak}}$ is high).
\[ C(0) = \sum_{\tau = \tau_{\text{peak}} - k \cdot (\frac{\Delta \tau}{2})}^{\tau_{\text{peak}} + k \cdot (\frac{\Delta \tau}{2})} C_{xy}(\tau) \quad \text{Eq. 12} \]

Where \( \tau_{\text{peak}} \) indicates the position of the maximum peak of the correlogram, \( \Delta \tau \) is the bin size of the correlogram function, and \( k \) is the number of bins around the peak. If \( C_{xy} = C_{yx} \) (i.e. autocorrelogram) the quantity \( C(0) \) and \( C_{\text{peak}} \) are equal, since the maximum peak of the autocorrelation function falls exactly in zero.

**Peak Latency**

The peak latency is calculated as the time of occurrence of the maximum peak of the correlogram and zero. It simply represents the time shift (or time lag) between two spike trains and it is usually represented in the form of an histogram.

*Figure 5: C(0), C_{\text{peak}} and peak latency on a cross-correlation sketch*
Coincidence Index – CI

To quantify the reproducibility of correlated spike timing it is used a measure called Coincidence Index (i.e. CI) (Juergens and Eckhorn, 1997; Jimbo et al., 1999; Tateno and Jimbo, 1999). For each electrode couple, the CI is calculated as the ratio of the integral of the cross-correlation function in a specified area around zero \( CI_{\text{zero}} \) or around the maximum peak \( CI_{\text{peak}} \) to the integral of the total area, according to the definitions below:

\[
CI_{\text{zero}} = \frac{\sum_{\tau = -k*\frac{\Delta \tau}{2}}^{k*\frac{\Delta \tau}{2}} C_{xy}(\tau)}{\sum_{\tau = -T}^{T} C_{xy}(\tau)} = \frac{C(0)}{\sum_{\tau = -T}^{T} C_{xy}(\tau)} \quad \text{Eq. 13}
\]

\[
CI_{\text{peak}} = \frac{\sum_{\tau = \tau_{\text{peak}} - k*\frac{\Delta \tau}{2}}^{\tau_{\text{peak}} + k*\frac{\Delta \tau}{2}} C_{xy}(\tau)}{\sum_{\tau = -T}^{T} C_{xy}(\tau)} = \frac{C_{\text{peak}}}{\sum_{\tau = -T}^{T} C_{xy}(\tau)} \quad \text{Eq. 14}
\]

where \( T \) is the correlogram window, \( \Delta \tau \) is the bin size of the correlogram function, \( k \) is the number of bins around zero or around the peak and \( \tau_{\text{peak}} \) indicates the position of the maximum peak of the correlogram. Defined in the previous way, the numerator of the fraction represents the cumulative correlogram within the \( k \) central bins \( CI_{\text{zero}} \) or the \( k \) bins around the peak \( CI_{\text{peak}} \) and the denominator represents the cumulative correlogram in the entire time window \([-T, +T]\).

Fold change

The extent of a change in a cross-correlation function can be quantified in terms of fold change of association strength \( xC(\tau) \), as reported in (Eytan et al., 2004). Fold change means that you take, for each activity pair, its \( C(\tau) \) value at one session (i.e. \( C_{\text{post}}(\tau) \)) and divide it by its \( C(\tau) \) value in the previous session (i.e. \( C_{\text{pre}}(\tau) \)). How much did it change between two sessions relative to its initial strength? Did it double, triple its strength or what?
This measure is preferred to the delta change (i.e. $\Delta C(\tau) = C_{\text{post}}(\tau) - C_{\text{pre}}(\tau)$) in order not to bias towards changes only in relatively strong pairs but to consider each pairs relative to its own strength. Thus for any two sessions you get the fold change of all activity pairs and can ask yourself how does their distribution look: to do this you have to look at the logarithm of the fold change distribution.

![Histogram of the distribution of changes in correlation, log ($xC(\tau)$).](image)

*Figure 6: Example of the histogram of the distribution of changes in correlation, log ($xC(\tau)$). In this case the distribution is centred around 0; this means that the most likely value of $xC(\tau)$ (i.e. the most likely ratio $\frac{C_{\text{post}}(\tau)}{C_{\text{pre}}(\tau)}$) is equal to 1.*

**Correlation Coefficient**

The correlation between two variables reflects the degree to which the variables are related. The most common measure of correlation is the *Pearson Product Moment Correlation* (called also *Pearson's correlation* or *correlation coefficient* or, simply, *correlation*). When measured in a population, the Pearson's correlation is designated by the Greek letter $\rho$. When computed in a sample, it is designated by the letter $r$ and is sometimes called *Pearson's r*. Pearson's correlation reflects the degree of linear relationship between two variables and it ranges from +1 to -1. A correlation of +1 means that there is a perfect positive linear relationship between variables (i.e. high scores on the X-axis are associated with high scores on the Y-axis).
A correlation of -1 means that there is a perfect negative linear relationship between variables (i.e. high scores on the X-axis are associated with low scores on the Y-axis). A correlation of 0 means there is no linear relationship between the two variables. Correlations are rarely if ever 0, 1, or -1. The correlation coefficient $r$ is calculated in the following way and it does not involve the previous computation of the cross-correlation function:

$$ r = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}} \quad \text{Eq. 15} $$

In the neuroscience field, the just presented linear correlation coefficient, calculated among different units of a neuronal network, a recently proposed synchronization measure (Selinger et al., 2004). In that paper, the problem has been specifically developed for series of time stamps, since the standard definition of correlation coefficient could not be applied by using binary data. In this case, it is much better to use the following formula which avoids to use the means and is therefore much faster to calculate:

$$ r = \frac{\sum_{i=1}^{N} x_i y_i - \frac{1}{N} \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{\sqrt{\sum_{i=1}^{N} x_i^2 - \frac{1}{N} (\sum_{i=1}^{N} x_i)^2} \sqrt{\sum_{i=1}^{N} y_i^2 - \frac{1}{N} (\sum_{i=1}^{N} y_i)^2}} \quad \text{Eq. 16} $$

In particular, if $x(t)$ and $y(t)$ are spike trains composed only by zeros and ones, it is possible to further reduce the complexity of the above formula considering that:

1. $\sum_{i=1}^{N} x_i = N_x$, meaning that the total number of spikes in the x train is $N_x$;
2. $\sum_{i=1}^{N} y_i = N_y$, meaning that the total number of spikes in the y train is $N_y$;

The Pearson’s coefficient can be easily calculated as follows:

$$ r = \frac{N \sum_{i=1}^{N} x_i y_i - N_x N_y}{\sqrt{N_x(N - N_x)}\sqrt{N_y(N - N_y)}} \quad \text{Eq. 17} $$
2.2 PARTIAL CORRELATION

As said before, the identification of the causal relationships between pairs of neurons is important in the study of synaptic interactions within the nervous system at population level. Functionally relevant neuronal connections can be defined and identified by changes in discharge probability in a postsynaptic neurons. While direct (i.e. monosynaptic) connections are devoid of interneurons, polysynaptic pathways involve one or more intercalated neurons. As explained before, the simplest approach to identify the association between neural spike trains, uses the cross-correlation histogram (Perkel et al, 1967) and its variants (Aersten and Gerstein, 1985; Eytan et al, 2004; Chiappalone et al, 2006; Le Feber et al, 2006). However, when analysing the structure of a larger neural ensemble, we cannot infer from the cross-correlation histogram to what extent these changes are due to a direct connection between the two neurons, or to common inputs; thus cross-correlation is not able to distinguish between direct and indirect connections (Eichler et al, 2003). This limitation was overcome with the notion of partial coherence (Brillinger et al, 1976), where, in assessing the dependence between two spike trains, the effects of the activity of all other spike trains, (assumed to be additive) have been removed (Rosenberg et al, 1989). Simulations showed that for up to 30 neurons, partial coherence seems to work well in distinguishing between direct and indirect connections. Eichler (Eichler et al, 2003) extended the partial coherence model to the time domain, in order to try to distinguish between excitatory and inhibitory connections, providing information on direction and type of the interaction; this method uses a Scaled Partial Covariance Density (SPCD) function. Let us consider a neuronal population V and two specific neurons \( x, y \in V \). One simple indicator of functional connectivity among \( x \) and \( y \) is the standard correlation \( R_{xy}(\tau) \) of their associated spike trains: \( x(t) \) and \( y(t) \). Its Fourier transform, the cross-spectral density \( |S_{xy}(\omega)|^2 \), can be used to estimate the spectral coherence \( |C_{xy}(\omega)|^2 \) where:

\[
C_{xy}(\omega) = \frac{S_{xy}(\omega)}{\sqrt{S_{xx}(\omega)S_{yy}(\omega)}}
\]

Eq. 18
and $S_{xx}(\omega)$ and $S_{yy}(\omega)$ are the power spectra of $x(t)$ and $y(t)$. Partialization (Brillinger et al., 1976) consists of subtracting from $S_{xy}(\omega)$ the effects of all (possibly multivariate) spike trains $z(t)$, where $Z = V - [x, y]$, so that:

$$S_{xy|Z}(\omega) = S_{xy}(\omega) - S_{xz}(\omega)S_{zz}(\omega)^{-1}S_{zy}(\omega) \quad \text{Eq. 19}$$

Correspondingly, its inverse Fourier transform, $R_{xy|Z}(\tau)$, is the partial covariance density and consequently, partial coherence function, $|C_{xy|Z}(\omega)|^2$ can be defined as well. It has been shown (Dahlhaus, 2000; Eichler et al, 2003) that the partial spectral coherence can be efficiently computed by inversion of the spectral matrix, $S(\omega)$ of the whole set of nodes.

If $G(\omega) = S(\omega)^{-1}$ is such an inverse, then:

$$S_{xx|V\backslash\{x\}}(\omega) = \frac{1}{G_{xx}(\omega)} \quad \text{Eq. 20}$$

$$S_{yy|V\backslash\{y\}}(\omega) = \frac{1}{G_{yy}(\omega)} \quad \text{Eq. 21}$$

$$C_{xy|Z}(\omega) = -\frac{G_{xy}(\omega)}{\sqrt{G_{xx}(\omega)G_{yy}(\omega)}} \quad \text{Eq. 22}$$

$$S_{xy|Z}(\omega) = \frac{C_{xy|Z}(\omega)}{1 - |C_{xy|Z}(\omega)|^2} \quad \text{Eq. 23}$$

And substituting equations 20, 21 and 22 into equation 23, we get:

$$S_{xy|Z}(\omega) = -\frac{G_{xy}(\omega)}{\sqrt{G_{xx}(\omega)G_{yy}(\omega)}} \frac{1}{1 - |C_{xy|Z}(\omega)|^2} \sqrt{S_{xx|V\backslash\{x\}}(\omega)S_{yy|V\backslash\{y\}}(\omega)} \quad \text{Eq. 24}$$

To assess functional connectivity we actually take a scaled version of $R_{xy|Z}(\tau)$ (i.e. the inverse Fourier transform of $S_{xy|Z}(\omega)$) defined as:
\[ s_{xy|Z}(\omega) = \frac{R_{xy|Z}(\tau)}{\sqrt{r_xr_y}} \]  \hspace{1cm} Eq. 25

Where \( r_x \) and \( r_y \) are the values of the maximum peak of the autocorrelation function, and \( s_{xy|Z}(\omega) \) is the so called **Scaled Partial Correlation Density (SPCD)**.

To summarize, partialization allows to delete from the standard cross-correlation the **effect of indirect connections** and that of **common input**. It also allows to distinguish direction and nature of interaction (excitatory, inhibitory). However, if two connections converge to the same node, after partialization the two input nodes become correlated (marrying- parents effect) (Eichler et al, 2003).

*Figure 7: Partial correlation between the spike trains generated by two simulated neurons. In the inset there is a zoom of the cross-correlation’s peak between -10 and +10 msec.*
2.3 TRANSFER ENTROPY

Transfer Entropy (TE) is an information theoretic measure which allows to extract causal relationships from time series (Lungarella et al, 2007). It shares some of the desired properties of the Mutual Information (MI), and it also takes into account the history and the dynamics of the peak trains. Differently from MI, TE is not symmetric with respect to the exchange of the variables \( X \) and \( Y \). Additionally, with respect to cross-correlation based methods, TE is sensitive to linear as well as nonlinear causal interactions (Gourèvitch et al, 2007).

Let us consider a spike train \( X \); we indicate with \( x_t \) the number of spikes falling in the time window \( t \) (\( t \) being discrete), then:

\[
x_t^m = (x_t, x_{t-1}, x_{t-2}, \ldots, x_{t-m+1})
\]

\( \text{Eq. 26} \)

is the spike count vector for the past \( m \) time windows. Considering a second spike train \( Y \) and its spike count vector \( y_t^m \), TE can be defined as:

\[
TE_{Y \rightarrow X} = \sum_{x_{t+1}^k, y_t^l} p(x_{t+1}^k | x_t^k, y_t^l) \log \frac{p(x_{t+1}^k | x_t^k, y_t^l)}{p(x_{t+1}^k | x_t^k)}
\]

\( \text{Eq. 27} \)

Mathematically, TE can be interpreted as a measure of the deviation from the generalized Markov property:

\[
p(x_{t+1}^k | x_t^k) = p(x_{t+1}^k | x_t^k, y_t^l)
\]

\( \text{Eq. 28} \)

Where \( p \) denotes the transition probabilities conditioned to the past \( k \) and \( l \) observations of the spike trains \( X \) and \( Y \), respectively. Low TE values indicate that \( y_t^l \) has no influence on the transition probabilities of the state of \( X \), so that the assumption of a Markov process holds. On the other hand, high TE values indicate that the spike train \( Y \) influence the response of \( X \).

TE can also be written as (Shahaf and Marom, 2001):

\[
TE_{Y \rightarrow X} = MI (x_{t+1}, (y_t^l, x_t^k)) - MI (x_{t+1}, x_t^k)
\]

\( \text{Eq. 29} \)

Equation 29 states that TE measures the gain in information of knowing the future and the past of \( x_t \), once \( y_t^l \) is known.
2.4 CELL CULTURE, EXPERIMENTAL SET-UP AND DATA ANALYSIS

Dissociated cortical neurons were extracted from rat embryos and plated on 60-channel MEAs precoated with adhesion promoting molecules (poly-D-lysine and laminin), at the final density of 5-8 x 10^4 cells/device, which means about 1200–1400 cells/mm². They were maintained in culture dishes, each containing 1 ml of nutrient medium (i.e. serumfree Neurobasal medium supplemented with B27 and Glutamax-I (Brewer et al, 1993)) and placed in a humidified incubator having an atmosphere of 5% CO₂ and 95% O₂ at 37°C. Under these environmental conditions, cortical neurons showed excellent growth and robust synaptic connections that allowed us to record spontaneous electrical activity from 7 days in vitro (DIV) up to 5–6 weeks in vitro (WIV). The network electrophysiological activity was recorded after the third-fourth WIV to allow the maturation of synaptic connections among the cells of the network. The experimental set-up was based on the MEA60 System (Multi Channel Systems, MCS, Reutlingen, Germany). The electrophysiological activity was recorded without any chemical or electrical stimulation (i.e., it was referred only to the spontaneous activity). The recorded signals ranged from random spike activity to more complicated and synchronized burst signals (see figure 8). Extracellular recorded signals were embedded in biological and thermal noise. These raw signals were recorded and sampled at 10 kHz, and data were then processed off-line by using custom software. Spiking and bursting activities were identified by using a spike detection algorithm (Maccione et al, 2009). The previously validated algorithm is based on a Differential Threshold (DT) for each channel, and it is used to discriminate population spike events. Briefly, after setting the threshold to 8 times the standard deviation of the biological noise (Jimbo et al, 1998; Shahaf and Marom, 2001), the algorithm considers a portion of the signal and looks for the Relative Maximum/Minimum whose peak-to-peak amplitude is above the defined threshold. Then, a candidate spike undergoes additional checks, such as the peak lifetime period (set as 2 ms) and the refractory period (set as 1 ms), in order to ensure the correct identification of a spike and its precise timing.
Figure 8: (A) dissociated cortical neurons coupled to a MEA. (B) Raster plot of the electrophysiological activity: each row corresponds to a recording site, and each small vertical line corresponds to a detected spike. (C) Electrophysiological activity recorded from one microelectrode (Adapted from Garofalo et al, 2009).

2.5 MICRO-ELECTRODES-ARRAYS (MEA)

Micro-Electrode Arrays (MEAs) (see figure 10) are a technique relatively new since the first reported micro-machined device adopted for the culture and the monitoring of cellular electrical activity was fabricated in the 1970s (Thomas et al, 1972). Nowadays large random networks of cortical neurons developing in vitro and chronically coupled to MEAs (Taketani and Baudry, 2006) represent a well established experimental model for studying the neuronal dynamics at the network level (Eytan and Marom, 2006), and for understanding the basic principles of information coding (Rieke et al, 1997), learning and memory (Marom and Shahaf, 2002). In addition, these model systems represent a good tradeoff between
controllability/observability and similarity to the *in vivo* nervous system and allow accessibility and extensive manipulation both from the chemical and the electrical point of view (*Raichman et al, 2008*). The use of MEA allows simultaneous recordings from hundreds of microelectrodes, giving the opportunity to access several "nodes" of the neural networks. From these recordings, it is possible to study how neurons are connected each other, and which topological architectures underlie a specific dynamic behaviour (*Teramae et al, 2007*). Within this topic, recent technological efforts (i.e. increasing the number of electrodes and the spatial resolution), allow to obtain a more precise mapping of the network, up to a possible identification of its anatomical connections (i.e. the set of physical or structural-synaptic connections linking neuronal units at a given time). In particular, the *active pixel sensor array* (*Berdondini et al, 2009*) (see figure 9) with 4096 electrodes and the *high-density CMOS array* (*Matsuda et al, 2013*) with 11,000 electrodes, are the latest and the most relevant *in vitro* micro-electrodes recording's technologies.

*Figure 9*: (A) Overview of the high resolution electrophysiological platform: the CMOS-MEA chip made up of 4096 Active Pixel Sensors (APS). (B) Two levels of magnification of the electrodes.
Figure 10: MEAs produced by different companies: (A) MEA 30/200 made by Multi Channel Systems for dissociated cultures, (B) MED60 made by Panasonic, (C) multi-well chip for pharmacological manipulations and (D) 3D electrodes for slice preparations, both made by Ayanda-Biosystems.

2.6 NETWORK MODEL DESCRIPTION

We implemented a neuronal network model mimicking the electrophysiological activity of cultured cortical neurons under spontaneous conditions. Following the approach proposed by Izhikevich (Izhikevich EM, 2003), we developed a neuronal network model made up of 60 spatially distributed and synaptically connected neurons. To test the developed algorithms we implemented different network configurations by increasing level of complexity and similarity with the biological networks. The results presented in this work are related to one specific configuration: we implemented a network model (called EI) significant from a physiological point of view, including excitatory and inhibitory connections. In this configuration, we considered two different types of neurons to model excitatory and inhibitory populations: the former type belongs to the family of regular spiking neurons, and the latter to the family of fast spiking neurons (Izhikevich EM 2003; Izhikevich EM 2004). Regular spiking neurons fire with a few spikes characterized by short ISI at the onset of an
input. Differently, fast spiking neurons exhibit periodic trains of action potentials at higher frequencies without adaptation. To preserve the main characteristics of the structure of the in vitro cortical neurons to 4:1 (Marom and Eytan, 2005; Marom and Shahaf 2002; Braitenberg and Schultz, 1991). These two families of neurons were connected following a random topology (Erdős and Rényi, 1959).

We considered an increasing average connection’s degree from 10 (low connected) to 60 (fully connected) neurons. The most relevant parameters relative to EI networks are summerized in Table 1.

Spontaneous activity was obtained by introducing a randomly distributed simulation reproducing the effect of fluctuation in the membrane potential (Buchmann and Schulten, 1987) due to the distributed background activity.

All the simulations, performed in Matlab environment, lasted 600 s at 0.1 ms integration time-step. The simulation output was then peak-detected by means of a simple hard-threshold algorithm.

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Synaptic weights (exc; inh)</th>
<th>Number of neurons (exc; inh)</th>
<th>Average # of outgoing connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>EI</td>
<td>4÷16; -4÷-16</td>
<td>48;12</td>
<td>10÷60</td>
</tr>
</tbody>
</table>

*Table 1*: the most relevant parameters relative to EI networks. Synaptic weights are chosen randomly, with a uniform distribution, in the reported interval. Positive (negative) weights correspond to an excitatory (inhibitory) synaptic connection.

2.7 VALIDATION PROCEDURE

The connectivity methods previously introduced are almost independent of the synaptic strength and the number of connections. However, in realistic networks and under increased connectivity conditions, the performance of the connectivity methods decreases. To validate and quantify the performance of these methods on simulated networks, we used the receiver operating characteristic (ROC) curve.
2.7.1 ROC CURVE

The Receiver Operating Characteristic (ROC) curve (see figure 12) is a well-known technique for visualizing, organizing, and selecting classifiers based on their performance. In this work, ROC curves were used to compare the performance of CC, PC and TE. To better appreciate the comparison among these methods, we reduced the ROC curve to a single scalar value (AUC) representing the obtained performance (Bradley 1997; Fawcett, 2006). Since AUC represents the area of a portion of the unit square, its value will be always between 0 and 1. However, since random guessing produces the diagonal line between (0, 0) and (1, 1), which has an area of 0.5, a classifier should have an AUC higher than 0.5 (good classifiers should have AUC values close to 1).

A ROC curve is obtained by comparing the (SWM) and the Thresholded Connectivity Matrix (TCM). For a given threshold, all TCM elements are considered as possible functional connections. If one of the TCM elements corresponds to an existing connection (a non-zero value in the SWM), it is considered as a True Positive (TP) and if the connection corresponds to a zero value in the SWM, then it is considered as a False Positive (FP). Furthermore, the TCM elements equal to zero either correspond to an existing connection (a non-zero value in the SWM), called False Negative (FN), or they correspond to a null element, called True Negative (TN).

Figure 11 b. shows the comparison between SWM (left) and CM (right) for the simple neuronal network of Figure 11 a.

By changing the threshold a variable number of TPs, FPs, TNs and FNs were obtained. Finally, they are reported on a two-dimensional plot by using the following definition of true positive rate (TPR) and the false positive rate (FPR) (see figure 13):

\[
TPR = \frac{TP}{TP + FN} \quad Eq.30
\]

\[
FPR = \frac{FP}{TN + FP} \quad Eq.31
\]
Figure 11: A small network consisting of four neurons is considered. (A) Network graph. The numbers indicated by the arrows are the synaptic weights. (B) SWM (left) and CM (right); CM was built using the cross-correlation.

Figure 12: Example of ROC curve. This curve is relative to a partial correlation analysis of a simulated network with 60 neurons; the average number of outgoing connections was 30.
Figure 13: Definition of TP_rate and FP_rate of a ROC curve for a generic binary classifier. The matrix represented in the figure is called confusion (or contingency) matrix of the classifier (Fawcett, 2004).

One of the major limitations of ROC curves is that they do not contain, explicitly, the information regarding the absolute number of TP and FP. In addition, good AUC values (> 0.7) can be associated to low ratios of TP vs. FP (TPs < FPs). On the other hand, in order to evaluate the connectivity method performances, also the variations of the TPs and FPs values, as a result of changing the threshold in the TCM, have to be taken into account. As an example, at the beginning of the thresholding procedure of a network characterized by \( N \) connections, the FNs are limited by the actual number of connections (FN < \( N \)), while the TN value is very high (TN >> \( N \)). Thus, by increasing the threshold, the FPR initially will remain close to zero (TN is high) but the TPR will increase more rapidly because of the limited number of FNs, according to Equations 26 and 27.

2.7.2 THRESHOLD SELECTION

A crucial step to represent the functional connectivity by means of connectivity map is the thresholding procedure. Once that a CM is obtained by the application of a statistical method, high and low values in the CM are expected to correspond to strong and weak connections. A procedure to select the strongest causal effects is necessary because a CC value (i.e. a \( C_{\text{peak}} \), see section 2.1.6) is computed for each electrode pair independently of the existence of a direct or indirect link, a simply random co-activation or a noisy link.

Since the connection strength should be proportional to the peak amplitudes, a threshold
value for creating the CM preserving the strongest links and discarding the noisy and the weakest ones is necessary.

When a functional connectivity method is applied to experimental data recorded by MEAs where the SWM is clearly unknown, several threshold choices can be done, since a standard procedure to select the threshold does not exist.

To identify a correct threshold we used 3 different methods. At first we considered the MCC (Matthews Correlation Coefficient, see section 2.7.2.1.1) curve; then we implemented a method based on the ACC (accuracy parameter, see section 2.7.2.1.2). Finally we used a heuristic approach to identify a reasonable threshold, determining for which threshold there is the major overlap between the SWM (Synaptic Weights Matrix) and the TCM (Thresholded Connectivity Matrix), in simulated networks.

2.7.2.1 USEFUL PARAMETERS IN THE CONTEXT OF A ROC ANALYSIS

There are several single-value measures to evaluate the accuracy of an inference in the context of a ROC analysis. In this work we used Matthews Correlation Coefficient (MCC) and the Accuracy coefficient (ACC).

2.7.2.1.1 MCC

MCC is a cross-correlation coefficient between 2 groups of binary data. It is defined as:

\[
MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}
\]  

\text{Eq. 32}

The MCC takes on value of 1 if the 2 classes analysed match perfectly, whereas random guess yields approximately 0. In order to identify the optimal threshold, we built a MCC vs threshold curve, by increasing the threshold and computing the relative MCC value. Then we extracted the maximum MCC value: we considered the corresponding threshold as optimal to handle the Connectivity Matrix (Kobayashi and Kitano, 2012).
Figure 14: MCC vs Threshold curve. In this example, we used a connectivity matrix obtained by a cross-correlation analysis of a simulated network made up of 60 neurons (the same network used for figure 12). The average number of outgoing connections was 30.

2.7.2.1.2 ACC

The ACC (accuracy) is a common metric that can be extracted from a ROC analysis. It is defined as (Fawcett, 2004):

\[ ACC = \frac{TP + TN}{P + N} , \quad P = TP + FP , \quad N = TN + FN \]  \hspace{1cm} Eq. 33

In order to identify the optimal threshold, we built an ACC vs Threshold curve, by increasing the threshold and computing the relative ACC value. Later, similarly to the case of MCC, we extracted the maximum ACC value: we considered the corresponding threshold as optimal to handle the Connectivity Matrix.
2.7.2.2 HEURISTIC PROCEDURE TO DETERMINE AN OPTIMAL THRESHOLD

As previously described, in order to evaluate the implemented algorithm’s performances and to correctly apply it to experimental data, we used simulated networks. In particular, we used a heuristic procedure to identify the optimal threshold to handle the connectivity matrix provided by the algorithm.

We focused on six possible thresholds defined as \( \mu + n \cdot \sigma \), where \( \mu \) and \( \sigma \) are the mean value and the standard deviation respectively, of the non-zero elements of the connectivity matrix, while \( n \) is an integer with \( n \in [0, 5] \). We analysed 6 sets of simulated networks, with an increasing average degree of connectivity from 10 to 60 (see section 2.4). For each degree of connectivity we averaged the results over 20 different simulations. In order to identify the functional connectivity and to build the CM for each simulation, we applied the aforementioned described algorithms; thus we obtained three connectivity matrix for each

\[ \text{ACC vs Threshold curve. In this example, we used a connectivity matrix obtained by a cross-correlation analysis of a simulated network made up of 60 neurons (the same network used for figure 12 and 13).} \]
simulation (one per method). Later we computed the mean CM for each method and for each degree of connectivity; in this way we obtained 6 CM (one per degree of connectivity) for every connectivity method previously described. Then, we handled every CM with the six different threshold first introduced. Thus, we obtained six different Thresholded Connectivity Matrix (TCM) (one per method and per degree of connectivity). Finally, we compared these TCM with the Synaptic Weight Matrix (SWM), and determine the optimal threshold that has to be used when processing and analysing experimental networks.

2.8 CONNECTIVITY MAP

We used the aforementioned functional connectivity methods to infer connectivity maps of neuronal networks. In this work, at first we focused on applying such methods to estimate the functional connectivity of simulated networks; in this way we could evaluate the performances of our algorithm and determine the most appropriate threshold to handle the obtained CM, in order to preserve the strongest links and discard the noisy and weakest (see threshold selection sec. 2.6). Later, we apply our algorithm to in vitro neuronal networks, and we used the threshold obtained in the previous step to build an experimental CM.

For each pair of neurons, the connectivity method provides an estimation of the connection strength (one for each direction). Thus, each method is associated to a matrix, the CM, whose elements (X, Y) correspond to the estimated connection strength between neuron X and Y. High and low values in the CM are expected to correspond to strong and weak connections. By using such approach, inhibitory connections could not be detected because they would be mixed with small connection values. By thresholding the CM (figure 16), it would be possible to filter out the noisy and non-causal values (because they are expected to be small). Anyhow, for each threshold value, a connectivity map is obtained. These maps, deduced by considering only the strongest CM values, display the links which should correspond to the strongest synaptic pathways.
If one build a CM using the cross-correlation method, it will not be possible to discriminate if an identified connection is due to a direct or an indirect link. However, if CM is obtained using the partial correlation algorithm, its values will be relative only to direct links (see partial correlation sec. 2.2) (figure 17).
2.9 DESCRIPTION OF THE IMPLEMENTED ALGORITHM

We chose to implement the algorithm using C# and Visual Studio C# express 2010 as programming language and environment respectively.

2.9.1 DATA ADAPTATION AND PROCESSING

The implemented algorithm’s aim was to execute a cross-correlation and partial correlation analysis between couples of electrodes to infer functional connectivity in neuronal networks. Its input data are the set of the spike trains recorded from each electrode. This stereotyped data are obtained from the raw data directly recorded in experimental networks (using a procedure described in section 2.4), or by using simulations (as described in section 2.6).

At first, to use these data in the C# application, we needed to find a specific procedure to convert them from matlab to text format. Thus we implemented a matlab function that wrote a text file formatted in this way: the first element was the total number of spikes recorded; the other elements were indexes indicating the sample corresponding to each recorded spike. Later, in the C# application, when reading these text files, the data were recoded in the original format: a sequence of length equal to total sample’s number, whose elements were 1 if there was a spike in the corresponding sample, and 0 otherwise. In a first version of the algorithm, we used a jagged array (i.e. a useful structure provided by C#, which allows to allocate n-dimensional vector by specifying only the first dimension) to represent each electrode’s data. However, the jagged array’s allocation is a relative slow procedure, thus we tried to find another solution; in particular we used a sparse matrix to codify each electrode’s data. We considered the sparse matrix’s definition provided by Math.Net Iridium and the C# Matrix Library (CSML). This was an optimal choice because, as explained before, the electrodes’ data contained a large number of zeros. In this way, we obtained a dual improvement: the maximum number of electrodes that could be processed by the algorithm passed from 60 to 120, and its computational time was approximately halved (e.g. it passed from 6 hour to 3 hour and 15 minutes for a simulated network with average degree of 30).
2.9.2 COMPUTING THE SPECTRUM OF CROSS-CORRELATION

First of all, it was necessary to exclude silent electrodes from the cross and partial correlation analysis; to determine which of the electrodes were significant we chose a minimum mean firing rate (defined through the graphic interface, see section 2.9.5; the chosen default value was 0.1 spike/s). Thus, in the first part of the algorithm, we computed the mfr of each electrode, and we considered for the analysis, only the electrodes with the mfr higher than the minimum one (i.e. the non-silent electrodes).

Then, the algorithm calculated all the parameters necessary to compute the cross-correlogram (see section 2.1). In order to compute the cross-correlogram between a pair of electrodes, we used a correlation window divided in bin (the correlation window and bin size were set in the initial graphic interface). This window flows on the two analysed electrodes’ spike trains, counting the number of spikes for each bin (we used a bin of 0.1 milliseconds in order to have a maximum of 1 spike per bin). We considered also that two consecutive correlation windows could overlap for a certain portion (the max overlap was set in the initial interface); thus, we determine the effective number of correlation windows necessary to cover all the spike trains without overlap, and we used the procedure previously described, for each of these non-overlapping windows. At this point, we had two vectors, for each couple of electrodes analysed, whose elements were the number of spikes per bin.

Then, we computed the Fast Fourier Transformation (FFT) of these two vectors; the FFT’s definition was provided by the online free library Math.Net Iridium. Later, we determined the power-spectrum in the specific correlation window analysed, by multiplying the spectrum of the vector relative to the reference electrode with the complex conjugate spectrum of the vector relative to the target electrode. The algorithm repeated these operations for all the non-overlapping correlation windows; thus, following the approach proposed by Eichler and Dahlhaus (Eichler et al, 2003), it determined the cross-spectrum by summing the power spectrum obtained for each correlation window. At this point, the algorithm saved the spectral power matrix (i.e. a matrix whose element $a_{ij}$ is the cross-spectrum between electrode $i$ and electrode $j$) in order to determine the Scaled Partial Covariance Density (SPCD). Finally the IFFT (Inverse Fourier Fast Transformation) of the cross-spectrum provided
the cross-correlation between each pair of electrodes. Actually we took a normalized version of the cross-correlation (see section 2.1.4); in particular, the algorithm determined and saved every auto-correlation’s peak (i.e. the correlation between an electrode and it self). Thus, the cross-correlation value between each pair of electrodes was divided for the square root product of the correspondent autocorrelation’s peaks.

2.9.3 Computing the Scaled Partial Covariance Density (SPCD)

The algorithm computed the SPCD by following the approach proposed by Eichler (see section 2.2). First of all, it was necessary to invert the spectral power matrix, and we chose to use the Moore-Penrose pseudo inverse based on the SVD decomposition (see section 2.9.7.2). In particular, we implemented a specific function to compute the Moore-Penrose pseudo inverse by faithfully following its mathematical definition (contained in a personal little library); while the SVD decomposition’s function was provided by the library Math.Net Iridium. Finally we computed the Partial Covariance Density (PCD), and we took a scaled version determined by dividing the partial covariance density relative to each pair of electrodes, for the square root of the product of the correspondent autocorrelation’s peaks (the same normalization procedure used for the cross-correlogram).

2.9.4 Output Processing

Every post-processing operation on the cross and partial correlation (e.g. the building of Connectivity Matrix, the representation of ROC curve, every evaluation procedure) were performed in matlab environment; thus, it was necessary to convert the text file provided as output by the implemented algorithm in matlab format. We implemented a specific matlab function which read the text file resulted from the algorithm and produce a correlation table for each electrode. In particular, the correlation table for one electrode is a cell (i.e. a vector of vector) whose element \( j \) is the cross-correlation (or partial correlation) vector with the electrode \( j \). In order to build this correlation table, we formatted the output of the C# algorithm in this way: there was one text file per connectivity method for each electrode; this
file contained the cross-correlation (or partial correlation) vector preceded by the index relative to each electrode for all of the electrodes of the analysis.

2.9.5 GRAPHIC INTERFACE

The possibility to create a specific graphic interface for an application is a powerful tool provided by C#. Thus, we implemented a graphic interface, in order to acquire all the necessary parameters to perform the algorithm (see figure 18).

Figure 18: Graphic Interface of the implemented algorithm. Using this interface, it is possible to set the main parameters necessary to perform the cross and partial correlation analysis; there is also a time indicator in order to determine the computational time requested by the algorithm.
The parameters settable in this interface are:

- number of electrodes to analyse -> with our algorithm it is possible to consider only a custom number of electrodes as well as all the electrodes contained in the folder, whose address is specified in the interface;
- peak train’s folder -> the address of the folder containing the text file converted in the previous step with the matlab function;
- Overlap, correlation window, bin, mean firing rate (see section 2.9.2);
- Sampling Frequency;

There are also two time indicators to read the computational time necessary to compute the cross-correlation and the Scaled Partial Covariance Density (SPCD, see section 2.2).
2.9.6 SCHEMATIC DESCRIPTION OF THE IMPLEMENTED ALGORITHM

Figure 19: Detailed and schematic description of the implemented algorithm.
2.9.7 APPENDIX: MATHEMATICAL TOOLS USED IN THE ALGORITHM

2.9.7.1 SVD DECOMPOSITION

The SVD decomposition was discovered independently by several scholars like Beltrami, Jordan, Sylvester and others, from 1870 until 1930 (Parodi, 2013). Any \( mxn \) matrix \( A \) can always be decomposed as:

\[
A = U \Sigma V^T
\]  \hspace{1cm} Eq. 34

- \( U \) is \( mxn \) and orthogonal (i.e. \( U^T = U^{-1} \)) and its columns are eigenvectors of \( UU^T \);
- \( V \) is \( nxm \) and orthogonal (i.e. \( V^T = V^{-1} \)) and its columns are eigenvectors of \( U^TU \);
- \( \Sigma \) is \( nxn \) and diagonal. Its main diagonal contains the singular values of \( A \):
  \[
  \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0; \ p = \min (m,n).
  \]

2.9.7.2 PSEUDO INVERSE MATRIX

Given any \( mxn \) matrix \( A \), one can define the Moore-Penrose pseudo inverse matrix \( A^+ \) as the \( mxn \) matrix such that:

- \( A \cdot A^+ \cdot A = A \);
- \( A^+ \cdot A \cdot A^+ = A^+ \);
- \( A \cdot A^+ \) is hermitian;
- \( A^+ \cdot A \) is hermitian;

A complex matrix is defined hermitian if and only if it coincides with its conjugate transposed, that is \( a_{i,j} = \overline{a_{j,i}} \ \forall \ i, j \in [m,n] \).

The pseudo inverse matrix can be directly and simply derived from the SVD decomposition: Given the \( mxn \) matrix \( A \) decomposed as showed in eq. 34, its pseudo inverse matrix \( A^+ \) can be defined as (Parodi, 2013):

\[
A^+ = V \Sigma^+ U^T
\]
\[ A = V \Sigma^+ U^T \]  
\textit{Eq. 35}

where \( \Sigma^+ \) is obtained from \( \Sigma \), substituting each singular value \( \sigma_i \neq 0 \) with \( \frac{1}{\sigma_i} \).

If \( A \) is a \( nxn \) not singular matrix (i.e. \( \text{det} (A) \neq 0 \)), you get that \( A^+ = A^{-1} \).

2.9.8 COMPUTATIONAL LIMITS OF THE ALGORITHM

The implemented algorithm had computational limits due to the nature of its input data and the execution of operations in the frequency domain. The first executed operation was the reconversion of the Electrodes’ data in the original format represented by a sequence of length equal to total sample’s number, whose elements were 1 if there was a spike in the corresponding sample, and 0 otherwise. We used simulated networks sampled with frequency of 10 kHz lasting 10 minutes, so we had a total of \( 6*10^6 \) samples per electrode, which meant \( 6*10^6 \) integers stored in the Random Access Memory (RAM) per electrode; each integer required 4 byte in the RAM, so we had \( 4*6*10^6 \) bytes, that is about 23 Megabytes per electrode, thus the algorithm required 1380 Megabytes to memorize the input data for a set of 60 electrodes. Anyway, in a second version of the algorithm, we improved the memory organization by using a sparse matrix to store the input data (see section 2.9.1).

Most of the data used in the algorithm in order to determine the cross and partial correlation between pairs of electrodes are of complex type. We used the Math.Net Iridium C# library to manage the complex data; this library offered not only the complex type definition, but also an almost complete set of functions usable on it (e.g. the FFT and the SVD for complex matrix). Each complex data required 16 byte in the RAM. The algorithm determined the partial correlation following the approach proposed by Eichler, by using the cross power spectral matrix (see section 2.2 and 2.10.3). The cross power spectral matrix is a 3D matrix of dimensions \([n,n,nfft]\) where \( n \) and \( nfft \) are the number of electrodes involved in the analysis and the number of points to compute the FFT respectively. Thus, considering 1000 electrodes and 150 msec as the correlation window’s length, we have \( 1000*1000*2048 \) complex elements for a total of \( 1000*1000*2048 *16 \) byte, that is about 31 Gigabyte.
So it is necessary to change approach in the memorising strategy of the algorithm (at least for the power spectral matrix) in order to extend the algorithm to a number of neurons higher than 120.
3. RESULTS AND DISCUSSION

3.1 THRESHOLD SELECTION

In order to identify the implemented algorithm’s performances and to correctly apply it to experimental data, we used at first simulated networks. In particular, we used a heuristic procedure in order to identify the optimal threshold to handle the connectivity matrix provided by the algorithm (see section 2.7.2.2). We focused on six possible thresholds defined as $\mu + n \cdot \sigma$, where $\mu$ and $\sigma$ are the mean value and the standard deviation of the connectivity matrix respectively, while $n$ is an integer with $n \in [0,5]$. At first, we built the morphologic versus degree histograms (see figure 20) for each connectivity method used in this work (i.e. CC, PC and TE), for each degree of connectivity and considering $n = 1,2,3$.

*Figure 20: Morphologic versus degree histogram determined for the model (first column) and for the three connectivity methods used in this work: CC, PC, TE (last three columns). The histograms were built for each degree of connectivity (i.e. from 10 to 60), varying the threshold $\mu \pm n \cdot \sigma$ ($n=1$ red, $n=2$ blue and $n=3$ green). Partial correlation and $n=2$ were the method and the threshold respectively with the best fit with the model.*
In these histograms we have the connection’s degree on the $x$-axis (as reconstructed by the connectivity methods) and the relative number of neurons on the $y$-axis (i.e. the number of neurons with a specific degree of connection). We observed that partial correlation showed the best fit with respect to cross-correlation and transfer entropy; in addition this fit was obtained only for $n = 2$ (i.e. threshold $= \mu + 2 \cdot \sigma$).

Another possible approach to identify the optimal threshold, is to compare the SWM (Synaptic Weight Matrix) to the TCM (Thresholded Connectivity Matrix) computed for each of the different threshold’s values and for each connectivity method previously introduced. Thus, we built a histogram, representing the percentage of overlap between the TCM and SWM in function of the threshold used to handle the connectivity matrix for CC, PC, TE, and for each average degree of connectivity (see figure 21). We observed that $n = 2$ was an optimal choice to obtain a maximum percentage of overlap for every degree of connection.

![Graphs showing overlap percentage for different thresholds](image)

Figure 21: Comparison between structural and functional connectivity, expressed as the percentage of overlap between the SWM (Synaptic Weight Matrix) and TCM (Thresholded Connectivity Matrix) in function of the threshold and for each average degree of connection (see section 2.7.2.2). $n=2$ seems to be the optimal choice to obtain the maximum overlap for the different average degree of connections considered (from 10 to 60).
So $\mu + 2 \cdot \sigma$ seemed to be a suitable threshold’s value to manage the connectivity matrix produced by the algorithm for experimental networks.

Later, we asked ourselves if using the optimal threshold determined in the previous steps to manage the connectivity matrix, provided a sufficient amount of information. In other words, we tried to understand if the information provided by the implemented algorithm (i.e. the number of connections correctly determined) when using the identified optimal threshold, were sufficient to try to reconstruct the network’s topology. We determined the number of neurons and the number of links correctly identified by CC, PC and TE in function to the threshold (i.e. in function of $n$). Thus, we built a neurons versus threshold histogram and a number of link versus threshold histogram for each value of $n$ and for each average degree of connectivity (see figures 22 and 23).

*Figure 22*: Number of neurons versus threshold histogram. These histograms were built for each average degree of connectivity and for each connectivity method. With $n=2$ you not only get the best fit with the model’s morphology (see figure 17), but also a sufficient number of neurons.
Figure 23: Number of links versus threshold histogram. These histograms were built for each average degree of connectivity and for each connectivity method. With $n=2$ you get the best trade-off between the fit with the model’s morphology (see figure 17), and the number of links identified. We concluded that by using a threshold of $\mu + 2 \cdot \sigma$ to handle the CM obtained from simulated data, you get the best trade-off between the overlap with the model’s morphology and the number of neurons and links correctly identified, so it is the optimal threshold that has to be used for CM built from experimental data.
3.1.1 HEURISTIC THRESHOLD VERSUS MCC/ACC THRESHOLD

We compared the threshold computed by using the heuristic procedure to the threshold determined by the MCC and the ACC curves (see section 2.7.2). In particular, we considered the average differences between the heuristic threshold and the threshold corresponding to the maximum ACC’s value (that is exactly the same of that corresponding to the maximum MCC’s value) for the partial correlation method in simulated networks, at the varying of the average degree of connectivity from 10 (neurons low connected) to 60 (neurons fully connected), and using 20 simulations per degree of connectivity.

![Figure 24: Average ACC curves for partial correlation at the varying of the average degree of connection from 10 to 60 (red curves); the black curves represent the differences between the heuristic computed thresholds and the thresholds deriving from the ACC/MCC analysis (see section 2.7.2). These differences increase with the average degree of connectivity; however, the thresholds produced by the two aforementioned procedure are approximately comparable.](image-url)
In order to appreciate the aforementioned differences, we plotted them on the partial correlation average ACC curves at the varying of the average degree of connectivity (see figure 24). Then, we considered the differences in percentiles between the thresholds provided by the heuristic procedure and those deriving by the ACC/MCC curves; thus, we built an average percentile’s difference versus average degree of connectivity histogram (see figure 25).

![Figure 25: Average Percentile’s difference between heuristic and MCC/ACC threshold versus Average degree of connection histogram; we used 20 simulations per degree of connection (variable from 10 to 60). The error bars represent the Standard Error of the Mean (SEM). We can see that the aforementioned differences increase with the degree of connectivity, indicating that the heuristic threshold is approximately comparable to that of the ACC/MCC (see also figure 24) and it is particularly adapt for simulated networks with low and medium degree of connectivity.](image)

We observed that the thresholds’ differences increased with the average degree of connectivity. However, the heuristic threshold was approximately comparable to that of the ACC/MCC and, in particular, this similarity was maximized for simulated networks with low and medium degree of connectivity.
3.2 ROC CURVES

We evaluated the average ROC curves (see section 2.7.1) relative to the application of each connectivity method used in this work (CC in blue, PC in red and TE in black) to simulated networks. In particular we used simulated networks with an average degree of connectivity variable from 10 (neurons low connected) to 60 (neurons fully connected) and we averaged the ROC curve of 20 different simulations (with the same SWM) for each degree of connectivity.

![ROC curves](image_url)

*Figure 25: Average ROC curves at the varying of the average degree of connection from 10 (neurons low connected) to 60 (neurons fully connected). These curves were averaged from 20 simulations for each degree of connectivity, and the error bar represent the Standard Error of the Mean (SEM).*

We observed that all these curves were widely higher than the bisector (i.e. the ROC curve correspondent to random guess); in addition, the partial correlation curve was above those
of cross-correlation and transfer entropy for every degree of connection.

However, to better appreciate the comparison among these methods, we reduced the ROC curve to a single scalar value (i.e. the Area Under Curve, AUC see section 2.7.1) which represents a direct indicator of their performances (Bradley 1997; Fawcett, 2006); thus, we built an AUC histogram at the varying of the average degree (see figure 26).

We observed again that Partial correlation’s AUC was higher than those of cross-correlation and transfer entropy for every degree of connectivity. Thus, we concluded that by varying the connectivity degree of each neurons from 10 to 60, partial correlation was the connectivity method that showed the best performances.

\[ Figure\ 26:\ \text{average\ AUC\ histogram\ at\ the\ varying\ of\ the\ average\ degree\ of\ connection\ from\ 10\ to\ 60\ (20\ simulations\ per\ degree\ of\ connectivity);\ the\ error\ bar\ represent\ the\ Standard\ Error\ of\ the\ Mean\ (SEM).\ Partial\ correlation\ showed\ the\ highest\ value\ of\ AUC\ for\ every\ degree\ of\ connectivity.\ The\ differences\ between\ the\ different\ methods’\ AUC\ were\ statistical\ significant\ with\ a\ p<0.001\ test\ (except\ for\ average\ degree = 10,\ where\ cross\ and\ partial\ correlation’s\ AUC\ were\ statistical\ different\ with\ p<0.01),\ as\ showed\ by\ a\ Kruskal\ Wallis\ non\ parametric\ test. \]
3.3 ACC CURVE

We evaluated the average ACC curves (see section 2.7.2.1.2) relative to the application of each connectivity method used in this work (CC in blue, PC in red and TE in black) to simulated networks. In particular we used simulated networks with an average degree of connectivity variable from 10 (neurons low connected) to 60 (neurons fully connected) and we averaged the ACC curve of 20 different simulations (with the same SWM) for each degree of connectivity.

*Figure 27*: Average ACC curve at the varying of the average degree of connection from 10 (neurons low connected) to 60 (neurons fully connected). These curves were averaged from 20 simulations for each degree of connectivity; the error bar represent the Standard Error of the Mean (SEM).
We chose to evaluate this parameter, because it is a direct indicator of the performances for each connectivity method used in this work to identify the functional connectivity. In particular, by definition, the ACC indicates the percentage of true connections identified by the algorithm with respect to the total (see section 2.7.2.1.2). In particular the maximum value of the ACC is very significant; this value indicates the best performances of the method analysed, that can be obtained in correspondence of the optimal threshold (i.e. the threshold relative to the maximum value of accuracy). For every degree of connectivity we observed that partial correlation showed the most regular trend and the highest values of accuracy. Cross-correlation showed the most irregular trend, in fact for average degree of 20, 30 the ACC curve was monotone; while for average degree of 40, the curve showed a maximum for the highest values of threshold (this imply that the thresholded connectivity matrix contains only a few connections). Finally Transfer Entropy is the first method to reach the maximum performances (at a lower values of threshold), but its maximum value of the ACC is lower than the partial correlation one. In addition, the threshold correspondent to the maximum value of ACC is comparable to the optimal value of threshold determined by using the heuristic procedure (see section 2.7.2.2) $\mu + 2 \cdot \sigma$ (where $\mu$ and $\sigma$ are the mean value and the standard deviation of the connectivity matrix respectively). Given the importance of the ACC curve’s peak, we built a histogram of the maximum value of ACC for each connectivity method, at the varying of the average degree of connection (see figure 28). It is evident that partial correlation’s ACC peak is higher than those of cross-correlation and transfer entropy for every degree of connection. We used a Kruskal-Wallis non parametric test and we determined that these differences were statistical significant with a p-value <0.001. Thus, we concluded that by varying the connectivity degree of each neurons from 10 to 60, partial correlation was the connectivity method with the best accuracy (and then best performances).
Figure 28: histogram of the average maximum value of ACC for each connectivity method, at the varying of the average degree of connection from 10 to 60 (20 simulations per degree of connectivity). The error bars represent the Standard Error of the Mean (SEM). Partial correlation showed the highest accuracy for each degree. The differences between the three methods were statistical significant with a p-value < 0.001 (as showed by a Kruskal-Wallis non parametric test).
3.4 MCC CURVE

We evaluated the average MCC curves (see section 2.7.2.1.1) relative to the application of each connectivity method used in this work (CC in blue, PC in red and TE in black) to simulated networks. In particular we used simulated networks with an average degree of connectivity variable from 10 (neurons low connected) to 60 (neurons fully connected) and we averaged the MCC curves of 20 different simulations (with the same SWM) for each degree of connectivity.

Figure 29: Average MCC curve at the varying of the average degree of connection from 10 (neurons low connected) to 60 (neurons fully connected); the error bar represent the Standard Error of the Mean (SEM). These curves were averaged from 20 simulations for each degree of connectivity.
We chose to consider this parameter, because it is a common metric used to evaluate the performances of a connectivity method (Kobayashi and Kitano, 2012). In particular the maximum value of MCC is significant in order to identify the optimal threshold’s value (see section 2.7.2) to handle the CM for experimental data.

![Histogram of the average maximum value of MCC for each connectivity method, at the varying of the average degree of connection from 10 to 60. The error bar represent the Standard Error of the Mean (SEM). Partial correlation showed the highest value of MCC for each degree. The differences between the three methods are statistical significant with a p-value < 0.001 (as showed by a Kruskal-Wallis non parametric test).](image)

We observed that the threshold’s value correspondent to the maximum value of MCC is exactly equal to the one obtained using the ACC curve; in addition it is comparable to the one obtained using the heuristic procedure (see section 2.7.2.2). The MCC (as well as the ACC) is
also an indicator of the performances of a connectivity method at the varying of the threshold; in particular, the maximum value of MCC is an indicator of the maximum performances of a specific method. Given the importance of the MCC curve’s peak, we built a histogram of the maximum value of MCC for each connectivity method, at the varying of the average degree of connection (see figure 30). It is evident that partial correlation’s MCC peak is higher than that of cross-correlation and transfer entropy for every degree of connection. We used a Kruskal-Wallis test and determine that these differences were statistical different with a \( p\)-value <0.001. Thus, we concluded that by varying the connectivity degree of each neurons from 10 to 60 and with respect to MCC, partial correlation was the connectivity method with the best performances.

4. FUTURE PERSPECTIVES

4.1 APPLICATION OF THE IMPLEMENTED ALGORITHM TO EXPERIMENTAL DATA

In the results (see section 3) we showed that Partial correlation is the connectivity method with the best performances when analysing simulated networks. In addition, we used a heuristic procedure in order to determine a reasonable threshold to handle the connectivity matrix provided by the algorithm for experimental data (see section 2.7.2.2). We found that \( \mu + 2 \cdot \sigma \) (where \( \mu \) and \( \sigma \) are the mean value and the standard deviation of the connectivity matrix respectively) was the best trade-off between the overlap with the model’s morphology and the number of neurons and links correctly identified in simulated networks (see section 3.1). Thus, partial correlation and \( \mu + 2 \cdot \sigma \) were the connectivity methods and the threshold respectively, that represented the most reasonable choice to analyse experimental networks. In this section, we show an example of application of our implemented algorithm with the aforementioned connectivity method and the threshold, to a cortical network at the second, the third and the fourth week of development (see figure 31).
Figure 31: Graph Connectivity reconstructed by the implemented algorithm. These graphs were obtained from an *in vitro* cortical network at the 2°, 3° and 4° week of development. We used partial correlation as connectivity method and $\mu + 2\sigma$ as threshold to obtain the TCM.

### 4.2 EXTENSION OF THE ALGORITHM TO A LARGER NUMBER OF NEURONS

One of the central concepts of contemporary neuroscience is the investigation of how computational properties emerge in neuronal population, and how information processing and transmission is related to topological properties of neuronal networks. Nowadays, several studies about the functional analysis of *in vitro* neural networks are largely available in literature, but they usually consider only 60 electrodes (because the well-established experimental set-up is based on the MEA60 System, Multi Channel Systems, MCS, Reutlingen, Germany). With networks of this size it is possible to determine and analyse some dynamic's properties (e.g. the existence of bursting or spiking activity), but it is really hard to understand their real functional and structural features. In fact, when considering a reduced number (i.e. tens) of nodes, it is almost impossible to distinguish between the different kind of topologies (i.e. random, small world and scale-free); this is possible, only when using neural networks (and related number of electrodes) with a dimension at least two or three orders of magnitude higher. Hence, to obtain realistic information and to characterize a real neural
network, an analysis of thousands of neurons (at least 1000-4000 cells) is necessary. These types of analysis become possible thanks to the development of new technologies (i.e. the active pixel sensor array of 4096 microelectrodes (Berdondini et al, 2009) and the high-density CMOS array (Matsuda et al, 2013) of 11,000 microelectrodes). We applied our algorithm only to network made up of 60 neurons, in fact with the current approach, there are several computational limits that prevent its application to network with more than 60 neurons (see section 2.9.8). Thus, extending the algorithm to a higher number of electrodes is a fundamental future perspective.
5. REFERENCES

24. Raichman N, Ben-Jacob E. Identifying repeating motifs in the activation of
41. Fawcett, T. ROC graphs: Notes and Practical Considerations for Researchers, March 16, 2004;
42. Parodi Mauro: Metodi matematici per l’ingegneria, 10/2013;
RINGRAZIAMENTI

Ringrazio il Prof. Sergio Martinoia per avermi dato la possibilità di lavorare e crescere accademicamente nel suo laboratorio;

Ringrazio il Prof. Paolo Massobrio per la pazienza, la cordialità e l’interesse con cui ha direttamente seguito il mio lavoro di tesi in tutte le fasi di realizzazione;

Ringrazio il Dott. Daniele Poli per l’assistenza e al tempo stesso l’amicizia e il supporto dimostratimi in tutto il mio percorso di tesi.

Ringrazio tutta la mia famiglia per avermi sostenuto nel percorso di studi, in particolare:

mio padre per avermi sempre spinto a estrarre il massimo da me stesso, per avermi sostenuto economicamente e moralmente e per aver creduto nelle mie capacità;

mia madre per aver sempre messo i miei interessi e il mio benessere davanti ai suoi, per aver investito su di me (economicamente e moralmente) e per aver rappresentato un sostegno stabile e assiduo, dandomi la forza di non mollare e di credere in me stesso, facendomi sentire costantemente speciale ed amato;

mia nonna Lina per aver rappresentato molto più di quanto una nonna possa essere, per essere stata una seconda madre e aver creduto in me al punto da sostenermi economicamente e moralmente per tutta la durata del corso di studi e avermi spinto a investire su di me stesso andando a studiare in una città lontana, un grazie speciale per aver finto di capire il programma dei corsi che qualche volta le raccontavo, solo per non farmi smentire di parlare;

i miei fratelli Niko e Eric per la simpatia con cui hanno ascoltato e seguito il mio percorso di studi, per il modo in cui mi sono accanto nella vita e mi sostengono pur essendo molto giovani;

mia zia Giusy, mio zio Maurizio e i miei cugini Paolo e Pamela, perché hanno rappresentato la mia seconda famiglia e hanno condiviso con me gioie e difficoltà nel corso di tutta la vita, trattandomi come figlio e fratello rispettivamente, non facendomi mancare attenzioni e affetto.

Un ringraziamento particolare e speciale alla mia fidanzata Giovanna, per avermi sostenuto nei momenti di difficoltà, per aver condiviso con me ogni momento di questo percorso (e non solo), per aver rappresentato un raggio di luce nelle giornate più buie e per la freschezza che dona alla mia vita.