POLITECNICO DI MILANO

Scuola di Ingegneria Industriale e dell'Informazione

Corso di Laurea Magistrale in Ingegneria Informatica

Dipartimento di Elettronica, Informazione e Bioingegneria

A pipeline for Fluorescence Brain Imaging using Dynamic Clustering and Classification

Relatore: Prof. Pier Luca LANZI

Correlatore: : Prof. Tanya BERGER-WOLF

Tesi di Laurea di:

Umberto DI FABRIZIO Matr. 836991

Anno Accademico 2016-2017

Dedicato ai miei nonni.

ACKNOWLEDGMENTS

Voglio ringraziare i miei relatori: Prof. Pier Luca Lanzi, Tanya Berger-Wolf e Robert V. Kenyon per il supporto, l'incoraggiamento, la fiducia e il pensiero critico nei confronti del mio lavoro che non sarebbe stato possibile senza la loro guida. Un ringraziamento speciale va al Prof. Daniel LLano la cui competenza in neuroscienze e abilitá nella ricerca sono state fondamentali durante questa ricerca. Voglio ringraziare i membri del Computational Population Lab non solo per i consigli e il brainstorming, ma molto di piú per essere brave persone. Ringrazio Claudia per essere sempre al mio fianco e condividere le nostre esperienze. Ringrazio mamma Paola, papá Giovanni, mia sorella Silvia, tutti coloro che festeggeranno con questo traguardo e quelli che avrebbero voluto.

UDF

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

CHAPTER **PAGE**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (continued)

FIGURE **PAGE**

LIST OF FIGURES (continued)

FIGURE **PAGE**

LIST OF ABBREVIATIONS

fMRI	Functional magnetic resonance imaging
MEG	Magnetoencephalography
EEG	Electroencephalogram
DTI	Diffusion tensor imaging
ARI	Adjusted Rand Index
PCA	Principal Components Analysis
CommDy	Dynamic Community Identification
UIC	University of Illinois at Chicago

ABSTRACT

In the recent years the research has seen an enormous effort in modeling, analyzing and describing brain dynamics. This work aims to give a contribute to this field, by validating a pipeline to identify dynamic communities in mouse brain imaging data. Understanding the dynamic aspect of brain functioning has been for long ignored both for computational reason and for the lack of algorithms able to perform such type of analysis. Identifying dynamic communities represents a fundamental step in characterizing the behavior of the brain, in order to understand functional groups of neurons and their interactions. After validating the robustness and significance of the pipeline, the statistics collected from the brains are used to train a machine learning algorithm which is able to classify a young brain from a old one with an accuracy of 92%. A biological hypothesis for this difference is formulated and tested through the injection of a drug that slows down synapses, and the new machine learning algorithm reaches an accuracy of 82% validating this hypothesis. This last results sheds lights on the possibility to use this pipeline as the source of input data for a classification algorithm that could differentiate between different types of brains as well as illnesses (e.g. Alzheimer) and at the same time offers insights on the reasons why a certain difference exists.

SOMMARIO

Negli ultimi anni la ricerca ha compiuto uno sforzo enorme nel modellare, analizzare e descrivere le dinamiche del cervello. Questo lavoro mira a dare un contributo in questo campo attraverso l'analisi e la validazione di una pipeline per identificare comunitá neuronali dinamiche partendo da immagini del cervello di topi. Capire la dinamicitá nel funzionamento del cervello é un aspetto che é stato ignorato per molto sia per motivi legati a limiti computazionali sia per la mancanza di algoritmi capaci di eseguire questo tipo di analisi. Identificare le comunitá dinamiche rappresenta un passo fondamentale nel caratterizzare il funzionamento del cervello, al fine di capire quali siano i gruppi funzionali di neuroni e quali le loro interazioni. Dopo aver validato la robustezza e la significativitá della pipeline, un algoritmo di machine learning é stato addestrato attraverso le statistiche ottenute delle comunitá dinamiche neuronali al fine di classificare cervelli giovani da cervelli anziani con una precisione del 92%. A seguito di tale classificazione un'ipotesi biologica per motivare tale differenza é stata formulata e testata attraverso l'iniezione di una droga che rallenta le sinapsi. In seguito l'algoritmo di machine learning é stato addestrato sul dataset dei topi drogati, ottenendo una precisione nella predizione del 84%, validando l'ipotesi per cui cervelli anziani sono caratterizzati da sinapsi lente. Quest'ultimo risultato apre la strada all'uso di questa pipeline al fine di distinguere tipi diversi di cervelli così come tipi di malattie (e.g. Alzheimer) e allo stesso tempo offre preziose intuizioni sulle ragioni delle differenze fra cervelli.

CHAPTER 1

INTRODUCTION

One of the most important and fascinating challenges of the last two decades has been the quest to model, describe and understand the human brain[1; 2]. Enormous efforts have been put to capture the brain functionality across several fields from neuroscience to psychopathology[3], cognitive science and computer science.

Despite this work, the real computation happening between our neurons is still largely unknown. Recently we achieved a stage of the human history characterized by two fundamental advantages: one is the possibility to observe the behavior of neurons at a fine grained level, being able to collect images of the brain in which pixels represent hundreds of neurons[4]; the other big advantage is the enormous computational power that we have access to and that can be used to process a huge amount of unstructured data to extract precious information.

There is a growing body of literature[5; 6], which has highlighted the importance of capturing the dynamic behavior and evolution of the human brain in order to understand to true relationships between brain areas. This represents a big step forward with respect to the classical static analysis of any brain data for several reasons: first of all, an aggregated and statical analysis of human brain activity will only take research so far in understanding the brain functionalities, moreover we know have enough power to gather and process data with a extremely fine grained spatial and time resolution, for all these reasons a dynamic based approach to study the human brain is advocated. Given the possibilities that we have and the challenging problem of

modeling the human brain, the aim of this research is the following:

Given brain images representing functional brain connectivity over time, validate the use of dynamic communities methods on the networks generated from brain imaging data. The final goal is the establishment of a robust framework able to extract useful biological insights from brain images for the neuroscience community. The longer term goal is to provide insights to the neuroscience community to help them formulate hypothesis on how the brain works. In particular the pipeline focuses on extracting data on how its substructures (modules) collaborate and synchronize under several conditions: drug use, aging, mental disorders, neurodegenerative diseases. In this work the impact of changing analytical parameters such as sliding window size, correlation threshold, static community clustering techniques, dynamic communities (CommDy[7]) costs, will be systematically explored to understand and quantify the impact on the final dynamic communities. Finally the accuracy of the framework with respect to identifying young vs old brains will be tested and analyzed.

1.1 Outline

The following work is going to be divided in several chapters

Chapter 2 describes the state of the art in brain imaging and analysis discussing shortcoming of several techniques, in particular it focuses on the use of dynamic community identification in this body of literature. Chapter 3 describes each phase of the algorithm pipeline in detail in order to introduce the overall approach. Chapter 4 deals with the exploratory analysis of the data and the choice of parameters for the correlation network. The following chapter 5 and 6 deal with the validation of two fundamental steps of the pipeline (i.e. static and dynamic

communities) and the discussion of their robustness. Finally, Chapter 8 summarizes the work and the result and Chapter 9 suggests future directions supported by this work.

CHAPTER 2

RELATED WORK

In the era of big data, networks have been shown to be a powerful and intuitive tool to model relationships and interactions among different subjects. Brain networks represent no exception to this trend, and find a particular interesting and immediate application of graph theory models and measures.

Networks have been used in many fields of biology, to model metabolic pathways[8], protein-protein interactions[9], regulatory gene networks[10], food webs[11], animal behavior[12][13], ecology[14], mind reading[15], brain networks[5],psychopathology[3] and social networks[16].

It is well established that brain connectivity can be mapped effectively and intuitively using networks [1; 2] showing the tight relationship between the brain structure and graph theory[17]. A half-decade long history of work has highlighted a number of network measures and techniques that can be applied to study brain networks[18; 5], those networks exhibit highly non random attributes[2] and can be studied through easily interpretable network metrics such as those reported by Betzel at al.[19]: small-worldness[20], hubs and cores[21; 22; 23], structural rich club[24], modular architecture[25; 26] and economic wiring[27; 28], in order to formulate more and more hypothesis about brain mechanism. The analysis of those network is not only limited to measures but can be extended by adapting classic data mining clustering algorithms to group nodes in networks as well as algorithms developed ad-hoc for networks. For instance the challenge of identifying communities in networks has attracted much attention [29] because

communities directly highlight the hierarchical organization of the brain and have an intuitive yet powerful interpretation. Nevertheless there has been a lack of computational methods for the identification of dynamic communities i.e. those communities which last over time: a first solution based on the stochastic block model and Hidden Markov Model (HMM) has been proposed recently in [6], however the model is still based on identifying static network communities and a pure concept of dynamic community is missing.

In this prospective, brain networks mapping techniques perform an extremely important role trying to map the entire functional and structural connectivity of the human brain. The Human Connectome Project (HCP)[30], started in 2009, constitute the first attempt in collecting high resolution data of human brain connections within and across individuals, moreover it provides a large public dataset to be used for exploring, comparing and extracting insight by the research community.

Thanks to the systematic mapping of neural connections with the improvement of technologies, there has been an increasing attention in exploiting the huge amount of data. Recently emerging evidence suggests the highly dynamic behavior of functional connections (for a detailed review see [31]), highlighting the non stationary trait of the connections over time[32], within periods ranging from 5 minutes of a fMRI session[33; 34] to the whole human lifespan[19].

From a comprehensive prospective, this path brings together the modeling of the brain as a network, with the inherently dynamic behavior of the brain, thus suggesting the use of dynamic networks to explore the functioning of the brain and pointing out the need for algorithms able to extract time-dependent measures.

Considering the long term high level goal of understanding brain operating principles, being able to capture high spatial and temporal resolution data is essential. Recently a great result has been published that maps the Fruit Fly's Brain Network in 3D[35], thus creating a great data source for researcher to investigate the structural network of the brain; the result as been achieved using a technique know as x-ray tomography, indeed there are several technique frequently used to map the brain.

The most widely used techniques for brain imaging are fMRI[36], MEG[37], EEG[38] and DTI [39]. Each of these methods has its own peculiarities and captures different features of the human brain dynamics, thus it is worth to investigate and validate previous results, integrate different data sources and gain further insights in the path to discover the mechanisms of the brain.

A recent technique note as flavoprotein autofluorescence has been developed, and it is known that synaptic activity is strongly coupled to flavoprotein signals[4][40]. This imaging technique has an high temporal and spatial resolution, infact in each image (172x130) a pixel contains about 100 neurons and images are collected at 70 frames/second. On the contrary fMRI, MEG and EEG have a much lower spatial resolution $1mm^3$ which contains a few million neurons.

fMRI is the most widely adopted technique, which has been used in the last two decades to study the brain, during this time several techniques and libraries to support and analyze the data have been developed, which have greatly favored the spread of this technique. Despite the widespread use, fMRI has still problems, infact as pointed out by Eklund et al. "Up to 70% of fMRI analyses produce at least one false positive, challenging the validity of over 40,000

studies"[41].

This works uses flavoprotein fluorescence to collect data, this is because not only it has a higher spatial and temporal resolution but given the end goal of understanding the human brain functionality, it is worth to explore and compare results from different techniques and collect insights which may remain hidden using other tools. Finally, the intent of the work is to establish a pipeline for the autofluorescence technique, to support the analysis of the data collected by researchers, in the same way fMRI libraries have helped shedding lights on the raw fMRI data.

CHAPTER 3

PIPELINE

3.1 Description

The purpose of this study is to analyze mouse brain imaging data and validate the inference process of time-consistent dynamic communities. Most importantly being this study intrinsically exploratory and given the complete novelty of network theory approach applied to fluorescence imaging, the work will focus on validating the steps of the pipeline evaluating the impact of the choice of parameters on the inferred communities.

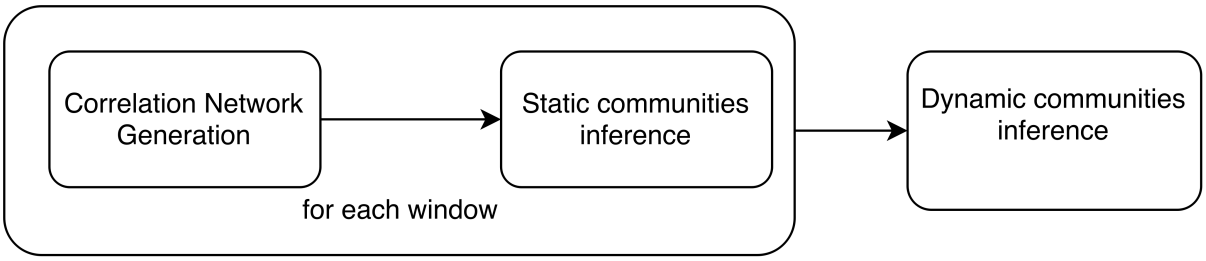


Figure 1: Pipeline flow to extract dynamic communities.

The process of inference of dynamic communities, as shown in Figure 1, involves three main steps which are detailed explained in the later paragraphs:

- Correlation network generation
- Static communities inference
- Dynamic communities inference

in a fourth additional step several statistics are collected from the inferred dynamic communities, which are in turn used for classification purposes. The reader can refer to Figure 2 as a guideline figure for all the steps of the pipeline.

3.1.1 Timeseries from Fluorescence Data

Flavoprotein autofluorescence imaging capitalizes on intrinsic fluorescence that occurs in mitochondrial flavoproteins as neurons activate and has been used across multiple preparations [52; 53; 54; 55]. This technique permits imaging of subthreshold activity across broad areas of the slice with high sensitivity [56; 57] and it is known that synaptic activity is strongly coupled to flavoprotein signals[4][40]. Images of the brain slices have a size of 172x130 pixels, each pixel is approximately 24 x 24 µm and represents about 100 neurons. From this images, the intensity of the pixels can be detected and this change in the pixel value can be interpreted as the firing of the underling group of neurons corresponding to a given pixel. The images are collected at $70 frames/sec$ for a total of a 1000 timestamps $\simeq 14 sec$ and by observing the variation of the pixel value during time we are able to capture the dynamicity of the signal traveling in the brain, this conversion is shown in Figure 3.

Each pixel in the 172 x 130 matrix is converted to a timeseries according to the above process. As shown in step 0 of Figure 2 (pixels are represented with a different color, just few are plotted for visual purposes) it is immediately clear that timeseries have the same trend

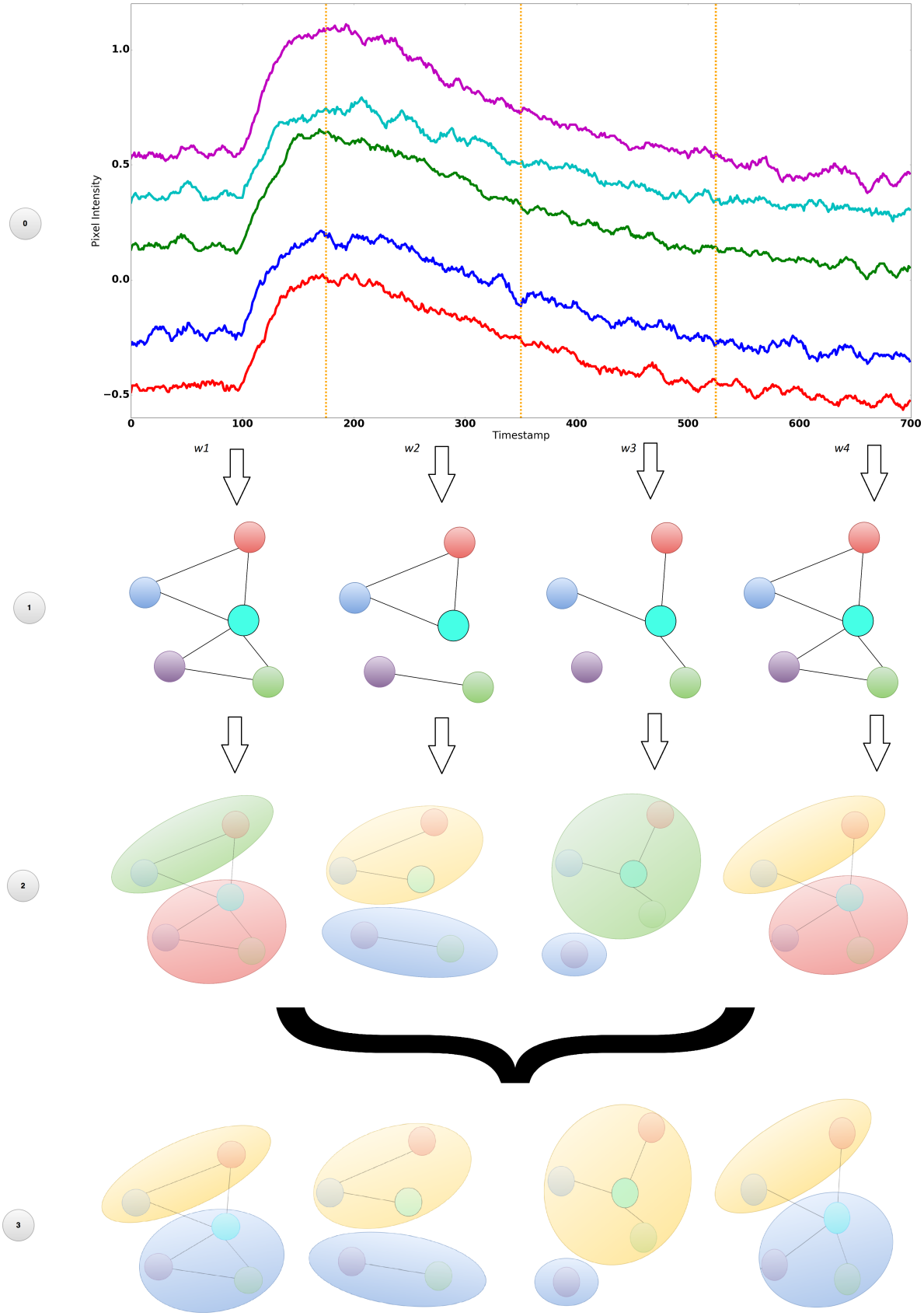


Figure 2: Pipeline flow to infer dynamic communities.0) The pixels intensity is collected during time to create timeseries.1)A sliding window process splits the timeseries in windows, a graph is generated in each window: nodes are pixel, edge represent the correlation. 2) The louvain algorithm identifies static communities is each network.3)Commdy infers dynamic communities.

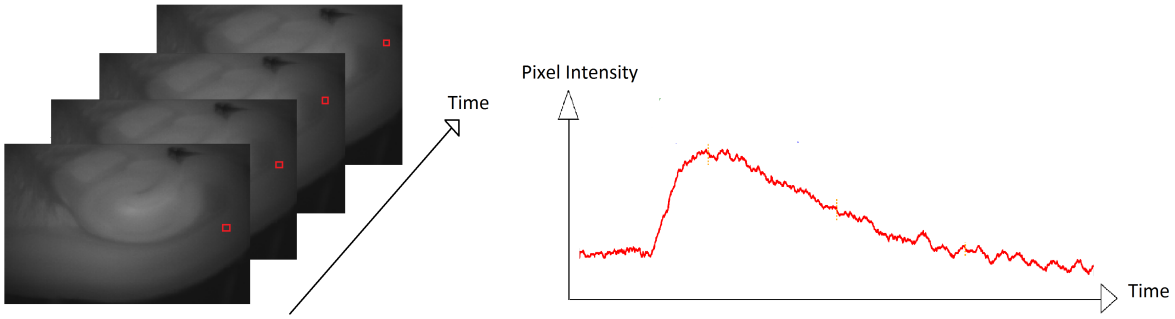


Figure 3: Conversion of pixel value from frames to timeseries.

which closely resembles the electric synaptic impulse, moreover the fact that the peak response occurs in the same timestamps, highlights the nature of the underling biological process: a seizure.

A sliding window process is used to grab the timeseries slices between a certain interval. In Figure 2 windows are non-overlapping for visual purposes, though the algorithm adopts a stride of 1 timestamp in order to capture the system evolution at a fine grained level.

3.1.2 Correlation Network

For each window, as described in the previous paragraph, the Pearson product-moment correlation coefficient between all pairs of pixel is calculated and if the correlation is higher than a threshold than an edge with weight equal to the correlation is put between the two nodes representing the two pixels. In so doing a thresholded correlation network is build, as shown in step 1 of Figure 2. In order to choose the threshold correlation coefficient and the window size a systematic variation of this parameters is performed as explained in chap-

ter 4 and the best values are selected to proceed to the next step of the pipeline. In this graph model, $G = (V, E, \omega)$ defines a graph with vertex set $V = \{v_1, ..., v_n\}$ and undirected edge set $E = \{e_1, ..., e_m\}$ that is augmented by a function $\omega(e)$ that gives a weight ($\in R$) to each edge $e \in E$. A weighted correlation network is generated with a list of weighted edges connecting pairs of nodes. By sliding the window 1 step each time over the entire timeline, a time series of correlation networks is obtained.

3.1.3 Louvain Algorithm

A community in a network is a collection of individuals (nodes) "among whom there are relatively strong, direct, intense, frequent, or positive ties" [51].

For each of the correlation network in the timeseries generated at the previous step, we then apply the Louvain static community inference method [42] to find groups of highly connected pixels: the "groups" that form the basis of CommDy. The purpose of this step is to find static communities of pixels in each correlation network, those communities can be interpreted as the group of pixel that in a given window are highly correlated to each other.

The Louvain algorithm was chosen because it is the leading (static) community detection algorithm and it is stable and scalable, moreover it closely capture the concept of community in the type of network that is generated, the reason being that Louvain is based on the number and weight of connection between nodes in contrast with other algorithms which are based on flow concepts. The algorithm is based on two steps that are repeated iteratively to optimize the modularity in the network. Once a local maximum of modularity has been attained, the algorithm rebuilds a new network where communities are the new nodes and edges are the sum

of weights of between all the nodes of these communities.The two steps are repeated iteratively, thereby leading to a hierarchical decomposition of the network, the process is iterated until convergence . In each static network, the nodes within the same cluster have relatively more edges connected to each other than to nodes outside the cluster.

3.1.4 Commdy Algorithm

The 3 step is the key computation of the pipeline. In this step the static communities, generated by Louvain in the previous step, are used to infer dynamic communities. In other words, at a very high level, we can see this step as a dynamic clustering method that groups nodes which are not necessarily connected in all the timestamp but which are found to belong more often than not to the same community along time. The CommDy algorithm (Dynamic Community Identification)[7; 64] was developed specifically to characterize dynamics, and was originally applied to the study of social networks, which has also been plagued by the same computational bottlenecks seen in brain imaging data. For example, CommDy has successfully characterized group behavior of Capuchin monkeys [58], leadership and following behavior of sheep [59], and equids [73], as well as social interactions among groups of people [7]. CommDy is peculiar because it explicitly allows for fluid community membership. CommDy basic idea is that functional groupings (clusters) tend to evolve moderately over time [61] and adopts a combinatorial optimization problem to detect them. CommDy input is a dynamic network, in the form of a time series of static networks. In each of these static networks, nodes have been grouped into communities which may come from observations, or inferred from the network using any of the static community inference methods[42; 49; 62]. The resulting optimization

problem is computationally intractable thus CommDy is an approximation algorithm with a provable approximation guarantee for dynamic communities inference [64]. CommDy takes as input the dynamic network with the inferred groupings at each time-step (using Louvain algorithm, in our case) and the three cost settings (for switching, visiting, and absence costs). However, it is the relative, rather than absolute, values of the costs that change the output structure found by CommDy. There are analytical limits on the range of the magnitude of the relative difference and the thresholds at which the underlying dynamic community structure changes. We will explore the effect of these cost settings on the resulting dynamic communities in the brain network.

3.1.5 Statistics from Commdy output

CommDy offers a quantitative description of community network dynamics . The simple variables quantify the distribution of size, number and duration of communities. In addition, we can identify the core versus peripheral members of communities by looking at the visiting and switching costs that individuals accumulate, i.e. an individual with a high switching cost is considered peripheral. Finally, we can use those metrics to compare community structures of different brains, the approach chosen in this case is a classification algorithm that learns the type of brain from those measures.

3.2 The Overall Picture

The pipeline presented in the previous paragraph is an attempt to capture dynamic communities in the brain. The contribution of this work is mainly a formalization and validation of these pipeline, whose results are validated at each step in order to ensure the stability of

the flow. Stability of the pipeline is evaluated with respect to the impact of a variation of the parameter on the final communities, moreover the significance of the results are discussed both quantitatively (e.g. accuracy of classifier) and qualitatively (feedback from experts).Finally the work is carried on relying only on the data and reducing the a priori knowledge of the dataset, this is in order to ensure the flexibility and extendability of the approach to other datasets.

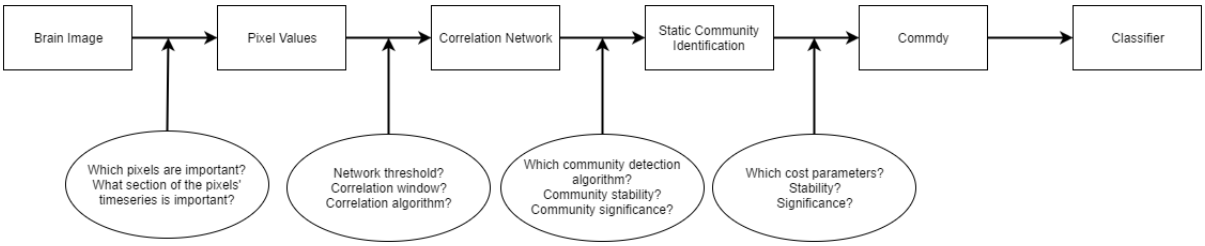


Figure 4: Pipeline critical choices.

As shown in Figure 4, each step of the pipeline presents several questions and issues to be resolved in order to test the functioning of the approach. Those steps and the issue connected with each one are the result of a thorough investigation and some problems were not foretasted since the beginning of the work. Anyway, the question that are going to be answered in this work are presented in this overview, to give the reader a big prospective on the direction and importance of the work, each of this issue will be analyzed and discussed in the next chapters

in greater detail.

- The first step is the choice of which pixel to consider as input and which part of the pixel timeseries to select. The approach of the work was not to make any a priori choice about the important pixels and section of pixels, in so doing all the results emerge from the raw and unbiased data giving more confidence in the robustness of the results. As a drawback the huge amount of data makes the pipeline slower.
- Once the timeseries of the pixel are chosen there are several choice to make in order to create the timeseries of correlation networks. First of all the correlation algorithm to be used (Pearson correlation, Dynamic Time Warping, Coherence), then the threshold used to discard noisy edges finally the correlation widow to be used.
- The correlation network is than analyzed to identify static communities, in this case the parameters are the community detection algorithm to use, the stability of the algorithm and the significance of the results.
- Finally, the identification of dynamic communities is based on a set of costs (switching, absence, visiting) whose impact has to be addressed with respect ot the stability and significance of the dynamic communities

CHAPTER 4

SLIDING WINDOW ANALYSIS

This chapter discussed the choice of the correlation algorithm, the threshold adopted to created the correlation network and the window used to calculate the correlation.

4.1 Correlation Algorithm

There are different correlation algorithm in the literature, among the most famous and the ones which are usually adopted in neuroscience related analysis are:

- Dynamic Time Warping (DTW)
- Pearson Product Moment Correlation

Dynamic Time Warping has been used extensively [68; 66; 67; 65] to find patterns in time-series and especially as a measure of similarity between timeseries. In general, DTW is a method that calculates an optimal match between two given sequences In order to determine a measure of the timeseires similarity, independent of non-linear variations, they are "warped" in time. This technique was particularly interesting for our case but the adoption of these technique has been rejected after a preliminary study on the dataset showing an unfeasible running time. Nonetheless, future works building on the results of this thesis may use the DTW by cleaning and reducing the dataset as explained in the conclusion chapter.

In this work Pearson product moment correlation has been chosen as the measure to eval-

uate the similarity between to timeseries because of it's wide use in the research community[19; 72; 32; 71; 6; 34] and given the short computing time needed.

Given two timeseries $x = x_1, x_2, ..., x_n$ and $y = y_1, y_2, ..., y_n$ the Pearson correlation is defined as:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

where $\bar{x}$ is the sample mean of timeseries x and $\bar{y}$ is the sample mean for timeseries y. Pearson correlation is also particularly suited because it captures the synchronous increment or decrement of two timeseries regardless of their mean.

4.2 Correlation Threshold and Window

As explained above, the Pearson correlation for a sample as to be calculated on a certain window, the resulting correlation value is then thresholded to avoid the introduction of noise effect in the correlation network. Given that there was no a priori rational to choose a window and a threshold over an other, in this work four main windows have been analyzed: 25, 50, 100 and 200 timestamps. For each window 20 correlation intervals have been created by dividing the correlation range from -1 to 1 into continuous intervals of 0.1 width.

The rationale of this preliminary analysis is that the best values for window and threshold are those which capture the pixels exactly in the auditory cortex. In order to capture this concept we decided to count how many correlations belong to a certain interval and in particular which area of the brain has more of those correlations.

The algorithm adopted is the following:

for each pair of pixel in a window the Pearson correlation is calculated and it is used to in-

crement the pixel counter of those pixels in the matrix map corresponding to that correlation interval. In other words, given that the algorithm adopts 20 intervals from -1 to 1 there are 20 correlation map initially empty, then the correlation between pixel X and Y is calculated, say 0.65 which belongs to the interval $[0.6, 0.7]$, so the cells (X,Y) and (Y,X) in the map corresponding to that interval are incremented.

This process (repeated for 5 young brains and 5 old brains) produces extremely matching correlation maps for all the samples, the results for each window are shown in Figure 5, Figure 6, Figure 7, Figure 8 where the heatmap scale goes from black (no correlations) to white.

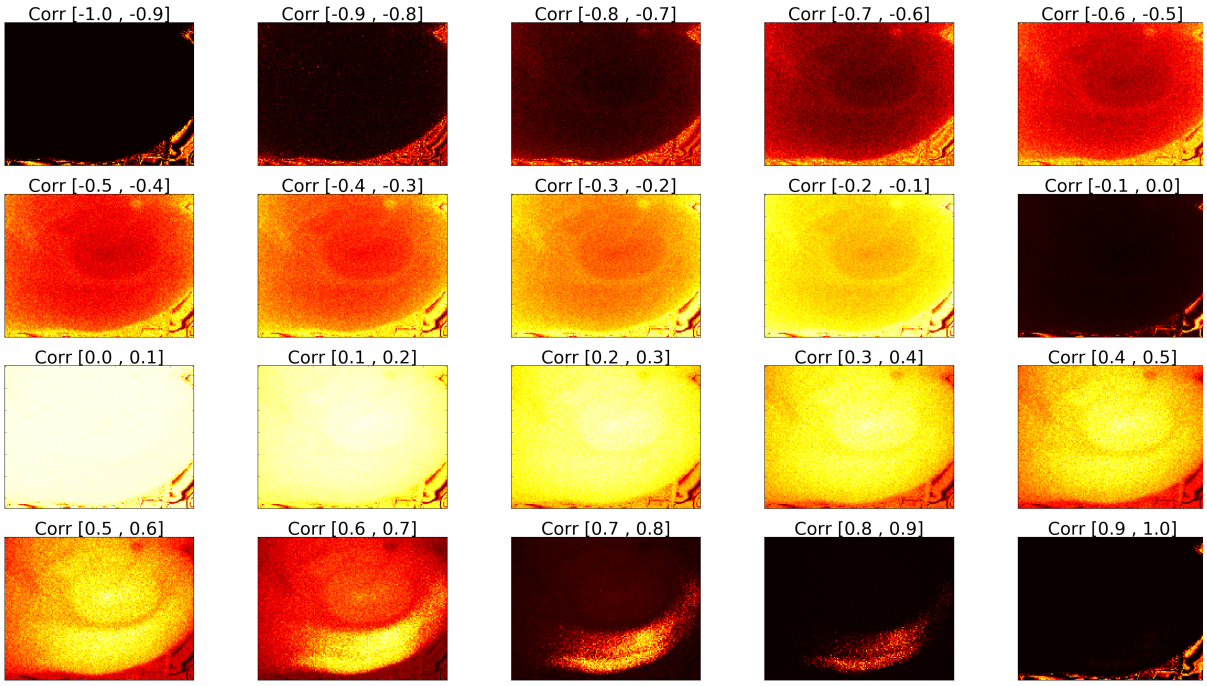


Figure 5: Correlation spatial maps for window = 25.

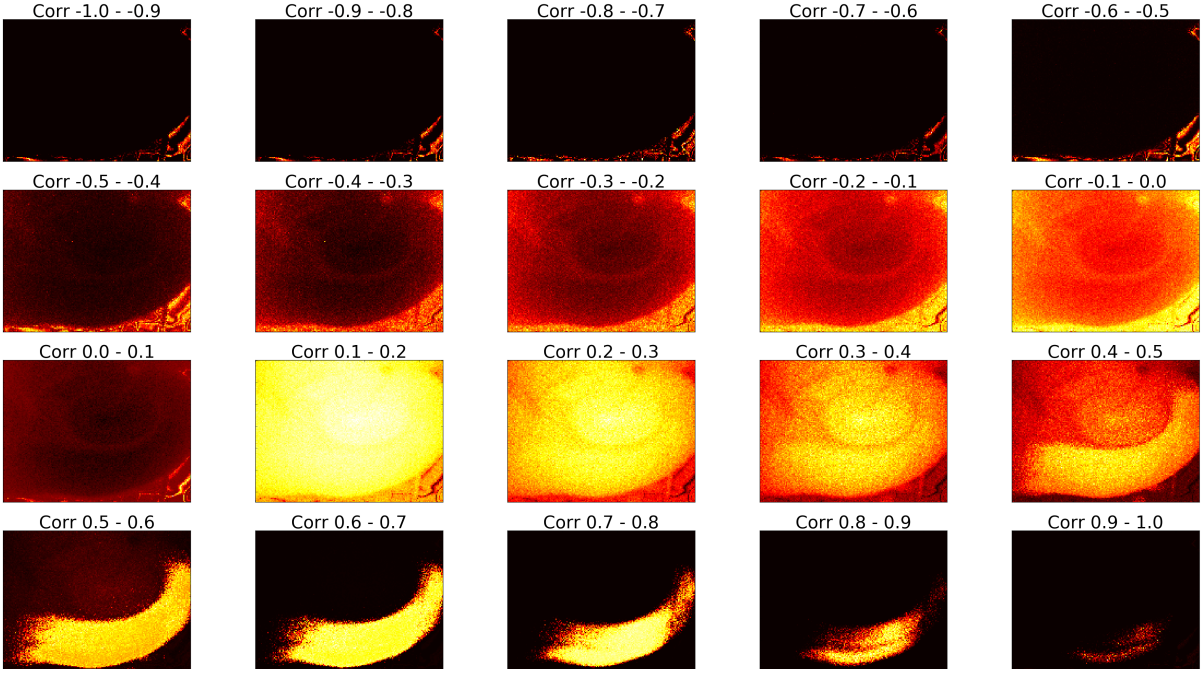


Figure 6: Correlation spatial maps for window = 50.

Figure 5 shows the spatial correlation maps when the window is 25 timestamps, it is possible to see that from correlation 0.4 we start to see the the shape of the auditory cortex although there is much noise.It is also noticeable (and reassuring) the fact that for negative values of the correlation nothing meaningful is captured. At a correlation value of 0.7 the shape highlights the auditory cortex but it's still quite noisy and the edge layer is not captured. The results for windows 50,100,200 are very similar, in these cases images quickly get much less noisy and the correlation value of 0.7 capture the shape of the auditory cortex clearly. Being the focus of the work on the dynamic aspect of the brain connections, the decision for the window and threshold

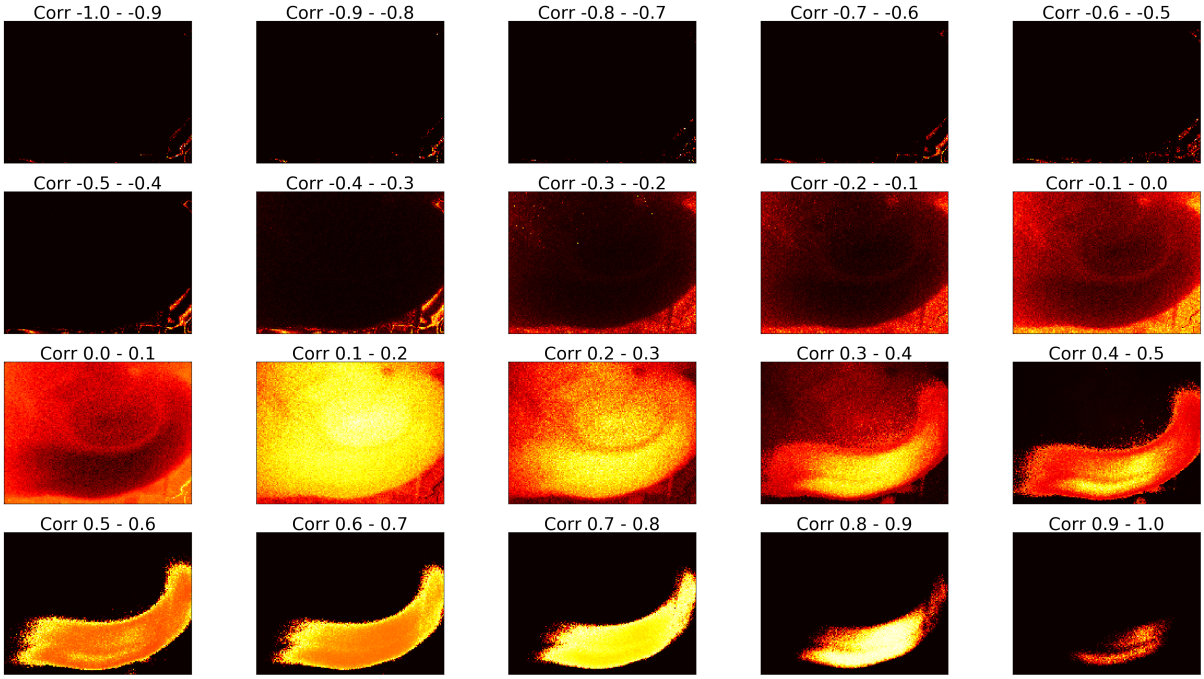


Figure 7: Correlation spatial maps for window = 100.

has been done by selecting the lowest window (higher time resolution) at the threshold which highlights best the auditory cortex without noise.**For these reasons the rest of the pipeline analysis has been done selecting a window of 50 and a threshold of 0.7**.

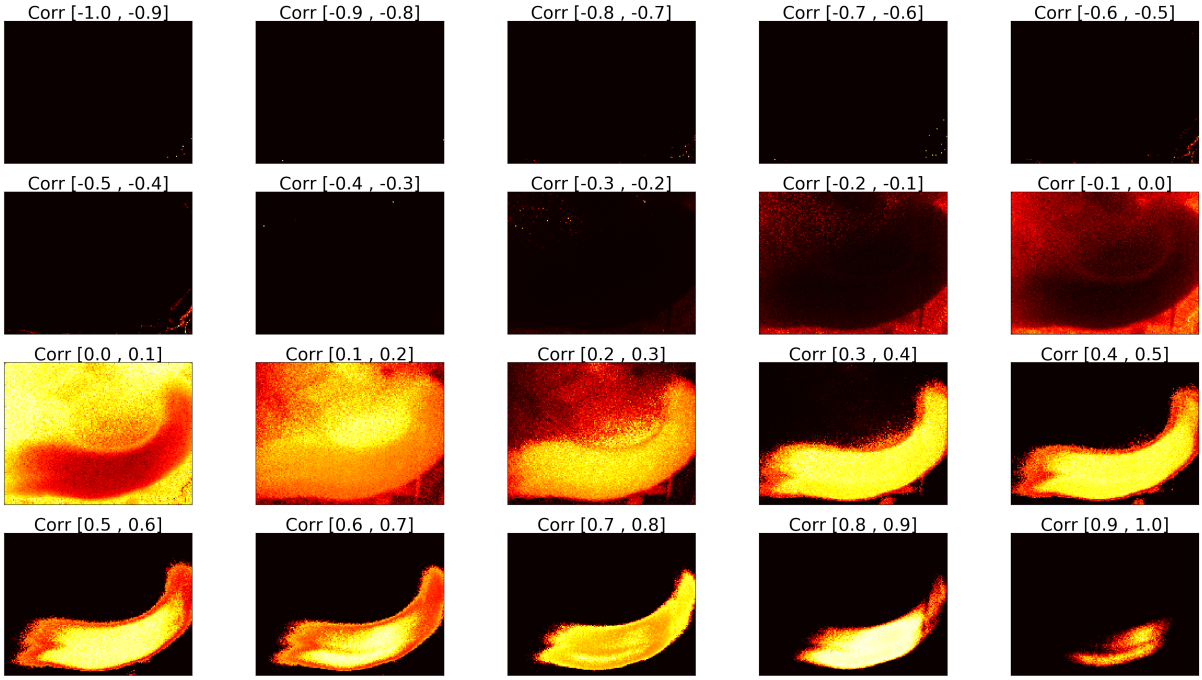


Figure 8: Correlation spatial maps for window = 200.

CHAPTER 5

LOUVAIN ALGORITHM ROBUSTNESS

5.0.1 What is Louvain

One of the most important challenges in networks is the identification of meaningful sub-structures to abstract from the hundred of thousand of nodes and be able to understand the high level functioning and organization hidden in the network structure. In order to obtain such information community detection (clustering) algorithms are used to identify groups on nodes (modules) that are strongly interconnected with respect to the rest of the network. Community detection algorithms are one of the most used and useful approch to the analysis of networks used by researcher, and there are three main types of approaches (for a complete review see [49]):

Null models Based on comparing the communities connectivity with respect to a proper null module. This is the direction of the algorithms based on modularity and the one exploited by Louvain which will be described later.

Block models Block models group nodes which are statistically equivalent in terms of connectivity intra- and inter- communities.

Flow models Based on flow rather than the topological structure. Communities consist of nodes among which the flow is more persistent once spreading in the group. This is the approach of infoMap that will be explained in chapter 5.2.

The Louvain algorithm has been chosen to perform the detection of static communities in the correlation networks. **Modularity(Q)** is the quantity that is optimized by the Louvain algorithm, a scalar in the range between -1 and 1 that measures static communities clusters by the density of edges inside them to edges outside them. Theoretically, optimizing Q, gives the best possible partitioning of nodes of a given network into communities.

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

where A_{ij} represents the edge weight between nodes i and j; k_i and k_j are the sum of the weights of the edges between nodes i and j, respectively; m is the sum of all of the edge weights in the graph; c_i and c_j are the communities of the nodes; and δ is a simple delta function.

Anyway there is no guarantee that the Louvain method produces stable and meaningful results on our dataset that is why a formal analysis has been carried on. Moreover Louvain is well known to be dependent on the order chosen to select the nodes,thus the objective of this chapter is to validate Louvain's communities.

This problem of order dependency was already addressed in [42], where the authors showed that the fluctuations of the modularity Q where negligible (mean=0.76 and std=10^{-2}) although a different order of nodes did cause a 7% variation of computation time. Anyway no formal guarantee on the stability of the algorithm has been given and the test were run on a single dataset (2.04 million nodes weighted network). For this reason the robustness of the Louvain

algorithm has been addressed on our dataset of Young and Old mice.

5.0.2 Modularity Stability

The first test on the robustness of the algorithm is run to prove that multiple runs of the algorithms on the same network give consistent results has regards the value of modularity. In order to perform this analysis 6 activations of young mice and 6 activations of old mice have been randomly chosen. For each of them the Louvain algorithm has been run 30 times.
As explained in Chapter 3, for one activation there are several correlation networks, one for each window. This approach empirically proves that by running the Louvain algorithm multiple times the Q value of the network in each time stamp does not vary significantly.
In order to do so the following algorithm is adopted:
for each activation a matrix is created as follows: the rows represent the windows and the columns the runs, a cell (i, j) represent the value of Q at window ith in the jth iteration, a sample of the data structure is shown in Fig. Figure 9.

The mean and standard deviation of modularity in each window is calculated, the plot in figure Figure 10 shows that the standard deviation is on the order of 10^{-3} which is less than the one reported in [42]. This may depend on the different scale of the network which is $\sim$8000 nodes for our dataset, not only this, but a little standard deviation supports the hypothesis about the existence of an underlying community structure in the brain processes. An other important observation can be made inspecting the graph: there are two obvious clusters of points, one from the lower left to the upper left and the other on the lower right of the figure. This indicates

	run0	run1	run10	run11	run12	run13	run14
0	0.888855	0.888855	0.888855	0.888855	0.888855	0.888855	0.888855
1	0.799945	0.799945	0.799945	0.799945	0.799945	0.799945	0.799945
2	0.833276	0.833276	0.833276	0.833276	0.833276	0.833276	0.833276
3	0.888779	0.888779	0.888779	0.888779	0.888779	0.888779	0.888779
4	0.888761	0.888761	0.888761	0.888761	0.888761	0.888761	0.888761
5	0.833300	0.833300	0.833300	0.833300	0.833300	0.833300	0.833300
6	0.899983	0.899983	0.899983	0.899983	0.899983	0.899983	0.899983
7	0.899973	0.899973	0.899973	0.899973	0.899973	0.899973	0.899973
8	0.929853	0.929853	0.929853	0.929853	0.929853	0.929853	0.929853
9	0.965954	0.965954	0.965954	0.965954	0.965954	0.965954	0.965954
10	0.960060	0.960060	0.960060	0.960060	0.960060	0.960060	0.960060
11	0.959389	0.959389	0.959389	0.959389	0.959389	0.959389	0.959389
12	0.927417	0.928699	0.928699	0.927080	0.928699	0.928699	0.928699
13	0.806904	0.809493	0.808673	0.809247	0.808398	0.809198	0.807928
14	0.689132	0.687636	0.691615	0.692551	0.686130	0.688736	0.689276
15	0.535289	0.534886	0.537195	0.533864	0.532106	0.535802	0.534952
16	0.379925	0.376950	0.382744	0.381339	0.380821	0.380912	0.371177
17	0.295867	0.298636	0.302680	0.300602	0.300218	0.300401	0.299292
18	0.248076	0.256361	0.246553	0.256981	0.251596	0.254531	0.248817
19	0.198297	0.197744	0.195693	0.200578	0.198481	0.198739	0.197150
20	0.174703	0.174787	0.170432	0.169915	0.173832	0.171974	0.167763
21	0.157790	0.155078	0.152290	0.158259	0.154720	0.159260	0.160005
22	0.137825	0.139342	0.135594	0.137866	0.135416	0.144296	0.138562
23	0.121955	0.118387	0.127226	0.114737	0.120580	0.121158	0.123971
24	0.114957	0.116254	0.110353	0.115244	0.115457	0.115948	0.112143
25	0.107602	0.101958	0.106621	0.104961	0.105362	0.103849	0.103023

Figure 9: Values of modularity for the correlation network build in each window. As appears from the pictures the modularity in a given timestamp does not change significantly. This is further confirmed by the mean and standard deviation analysis.

that networks with a strong modularity ∼1 usually have a small standard deviation because the algorithm may have effectively captured the real sub structures of the network, conversely when the modularity value is low ∼0 the algorithm fails to identify any strong community structure thus multiple runs of the algorithm have a higher standard deviation.

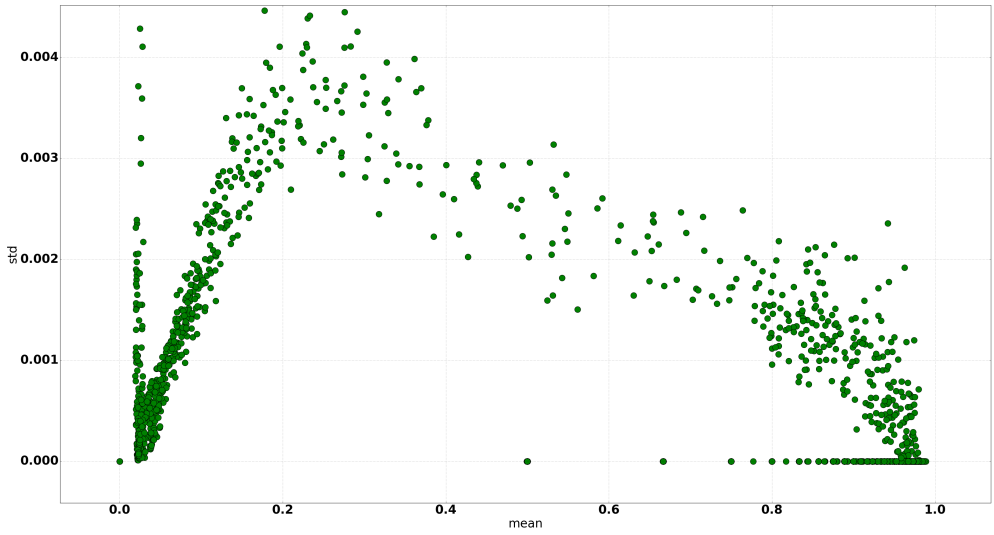


Figure 10: Each data-point represents mean and standard deviation of modularity (Q) for a given window. This plot aggregates several activations.

The question that arise is what is the cause of the 0 modularity mean. One may hypothesize a dependency on the timestamps. Figure 11 shows the plot of the mean value of Q with respect to the timestamp in which is calculated. The pattern is consistent among all the 12 activations

considered. The bottom line is that when the brain fires and most of the pixel light up the correlation network becomes big and without clear communities so Louvain cannot identify them precisely. This suggests that the real dependency is not in certain timestamps but

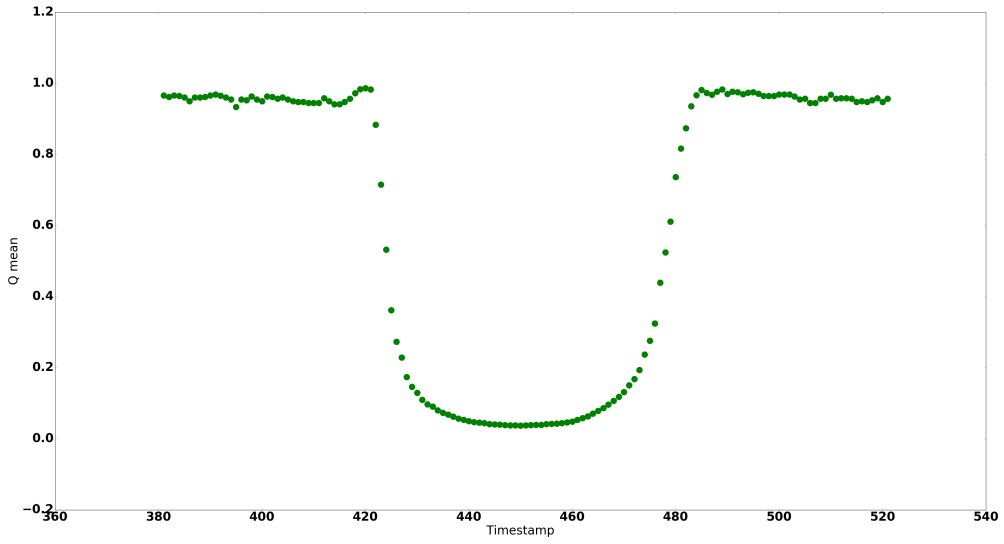


Figure 11: Each data-point represents mean of modularity (Q) for a given timestamp. The plot represent a single activation.

in the number of nodes of the network.The high correlation of the mean with respect to the number of nodes in the network is shown in Figure 12. In this case the plot clearly identifies the issue of the algorithm that struggles to identify any community in large networks. Anyway, networks are complex systems and one can also think that the dependency is not in the number

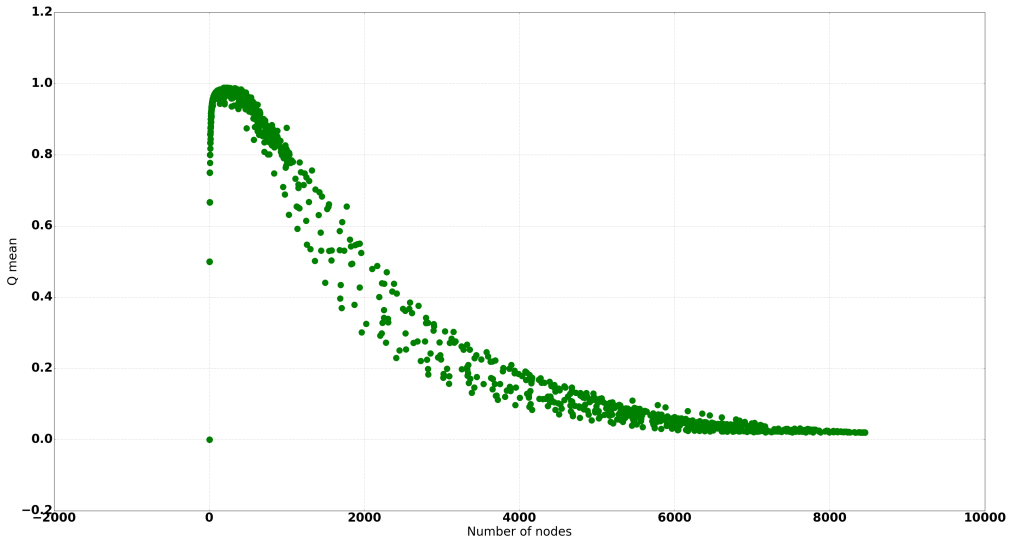


Figure 12: Each data-point represents mean of modularity (Q) as the number of nodes in the network vary. This is an aggregated plot of all the 12 activations.

of nodes but in the distribution of weights of large networks, Figure 13 shows the Q values as a function of some standard metrics of a network namely: the mean of the weights, the standard deviation of weights, the number of nodes and the number of edges . There is a clear correlation with the number of nodes, moreover when the mean of the weights of the network is close to 0.72 the modularity is high, and that mean is observed for networks before and after an activation.

The conclusion is that the Louvain algorithm is stable (same Q multiple runs) but it loosely captures the communities in the large brain networks, specifically it effectively captures communities pre and post activation whilst it fails to detect communities during the activation.

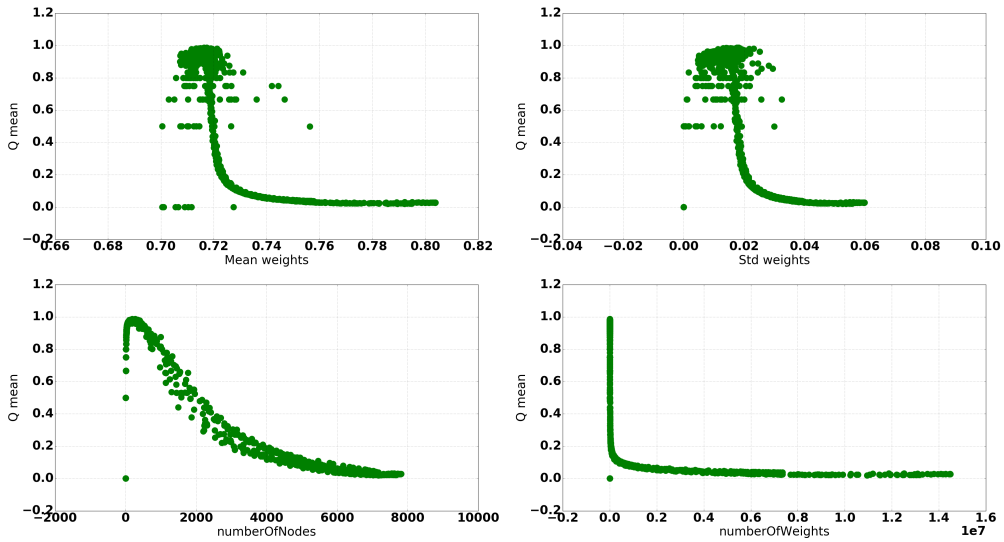


Figure 13: Dependency of Q value from the main four parameters of static networks. Each data-point aggregates 30 runs of Louvain on the same timestamp.

5.0.3 Structure stability

Although the modularity value is low but stable this does not say anything about the structure of the communities. This means that two networks may have the same Q but a completely different community structure or a low Q and have the same community structure, for this reason a way to address structure similarity is needed. There are tens of techniques to measure the similarity between partitions (for a in depth analysis see [43]),int this work the method that has been chosen is the Adjusted Rand Index because of its intuitive application to the problem and wide usage.

In order to define a measure to compare clusters we first need to define what a cluster is: A clustering C is a partition $C = \{C_1, C_2, ..., C_k\}$ of a set of data-points D such that $\bigcup_i C_i = D \wedge C_i \bigcap C_j = \emptyset, \forall(i,j), i \neq j$, thus it is a collection of disjoint subsets that cover all the points. One of the classical method to compare two clusters is by counting pairs. This is usually achieved by defining four variables:

- N_{11} The number of pairs of points that are in the same cluster both in C and C'
- N_{00} The number of pairs in different clusters both in C and C'
- N_{10} The number of pairs in the same cluster in C but in different clusters in C'
- N_{01} The number of pairs in different clusters in C but in the same cluster in C'

With those definitions the Rand Index(RI) [45] is an intuitive way to measure the similarity:

$$RI = \frac{N_{00} + N_{11}}{\binom{n}{2}}$$

where $\binom{n}{2}$ is the total number of possible pairs. In our case this measure is sound because we want to know if to partitions are made by cluster that contain the same points. The problem with the RI is that is does not account for chance, so we may obtain a score that is strongly depends on the chance of two partitions to be similar. This issue is address by using the Adjusted Rand Index (ARI) [44] which does account for chance, anyway as any method the ARI is not perfect because, as noted in [46], the baseline for two random partitions varies between 0.5 and 0.95. The ARI is a value between -1 and 1, it is close to 0 when the two partitions are randomly created and close to 1 when the two clustering are identical.

5.0.3.1 Method

The main problem is that this measure is a pairwise comparison, thus there is no direct application to multiple partitions comparison. In our specific situation, the objective is to compare multiple partitions (30), the chosen approach is to computer the adjusted rand index for all pairs of partitions.

In this way, given the network in one window and all the partitioning of that network among 30 runs, a matrix is build where each cell (i, j) is the Adjusted Rand Index between run_i and run_j for that window.Of course this matrix is symmetric and the diagonal is made of 1s. What is meaningful is the distribution of indexes, so that if most of the indexes are 1 we can say the

most of the partitions agree about the community structure in a certain timestamp.

Actually not only we care about consistency of partitions in a certain timestamp but also over time, thus if previously there was a 2D matrix of values for a certain window we will now have a 3D matrix where each 2D matrix represent a certain window. For each window the community structure has to be stable and this has to be true for each and every window, thus we can actually do a much more comprehensive analysis.

First we can look at the distribution of indexes overall the 3D matrix, this gives an overall idea about the distribution of values collected when calculating the index among runs and timestamps, then we look at the distribution of ARIs averaging all the values in a timestamp. The result of the overall distribution, shown in Figure 14, is very interesting: around 65% of indexes are 1s which means that there is an high degree of similarity and around 25% of indexes are actually close to 0 which means that the partitions completely disagree, finally a very small percentage of indexes falls between 0.1 and 0.9. The mean of this values is 0.7 and the standard deviation is 0.42 which is of course high on a scale between 0 and 1, moreover it has to be noticed that a mean of 0.7 is high (on the 0-1 scale) but by looking at the distribution we understand that this comes from a completely uneven distribution thus it's meaningless. This analysis seems to suggest that there are community structures that are consistently found and other that are totally random. The question that comes natural by looking at this figure is: who are the 0 values? Do this values all come from a set of timestamps in which Louvain fails to find the structure (and if so why) or they can be found in several timestamps, which suggest that in most timestamps there might by some runs that completely disagree

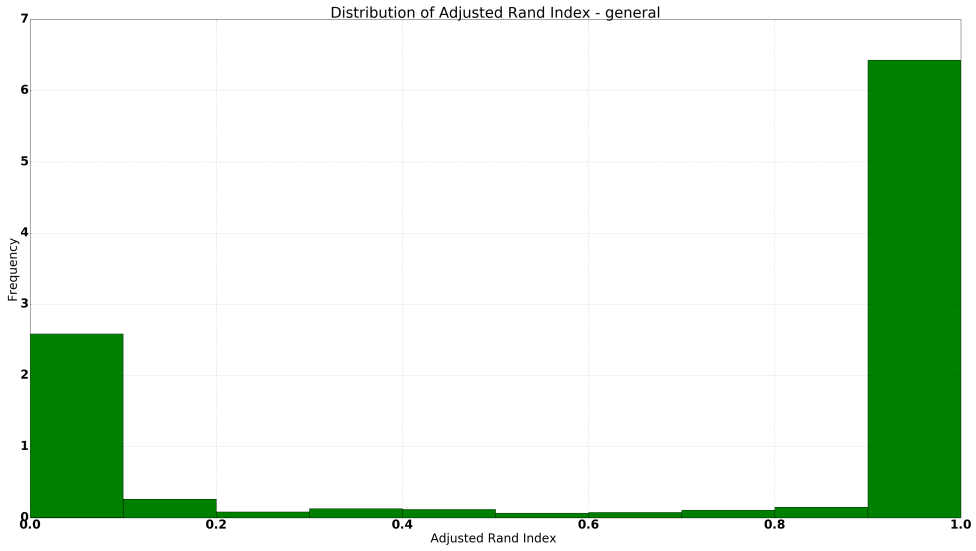


Figure 14: Distribution of Adjusted Rand Index over all pairwise comparisons.

with each other?. Of course the first hypothesis is the preferred one because we can say that except for those 'difficult' timestamps (it has to be understood why those timestamps exist) the Louvain algorithm is structurally robust. In order to address this issue the average of indexes in the 2D matrix in a certain window is computed and the distribution of this average among all the timestamps along with the value of the standard deviation for each timestamp, the result is shown in Figure 15. Figure 15 shows that the distribution for values for each timestamp is totally similar to the overall distribution, this supports the idea that in some timestamps Louvain completely fails to identify a consistent structure, infact the standard deviation calculated per timestamp is $\in [0 - 0.1]$ thus extremely regular for a timestamp.

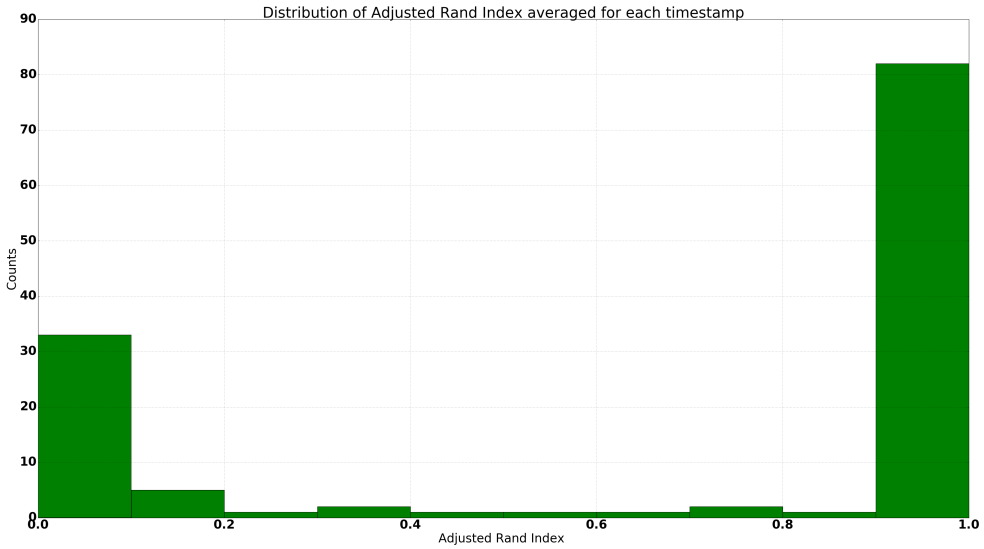


Figure 15: Distribution of Adjusted Rand Index averaging the indexes of the pairwise comparisons in each timestamp.

The question now is why there is such a high percentage of values close to 0? What are the characteristic of these network compared to the network with values close to 1?

5.0.4 Modularity vs Structure Robustness

The modularity robustness analysis suggests that, although the Q is stable there is an high percentage of values close to 0.

The structure analysis suggests that there is an high percentage of timestamps that agree with the structure of the partitions and also a non-negligible percentage in which runs totally disagree on the structure.

What is the relationship between the ARI index and the Q value?

Figure 11 suggests that the Q value is high in the pre and post activation, we would also like that in those timestamp the ARI would be close to one. During the core of activation the Q value is low thus we would expect also the structure similarity to be low. The comparison between ARI and Q among timestamps is shown in Figure 16, as expected the pre and post activation have both an high ARI and Q value which gives reliability on the clusters found. The activation core instead shows that the structure similarity somewhat rises after falling to 0. The meaning is that although Louvain fails to detect meaningful communities, those are consistent in structure.

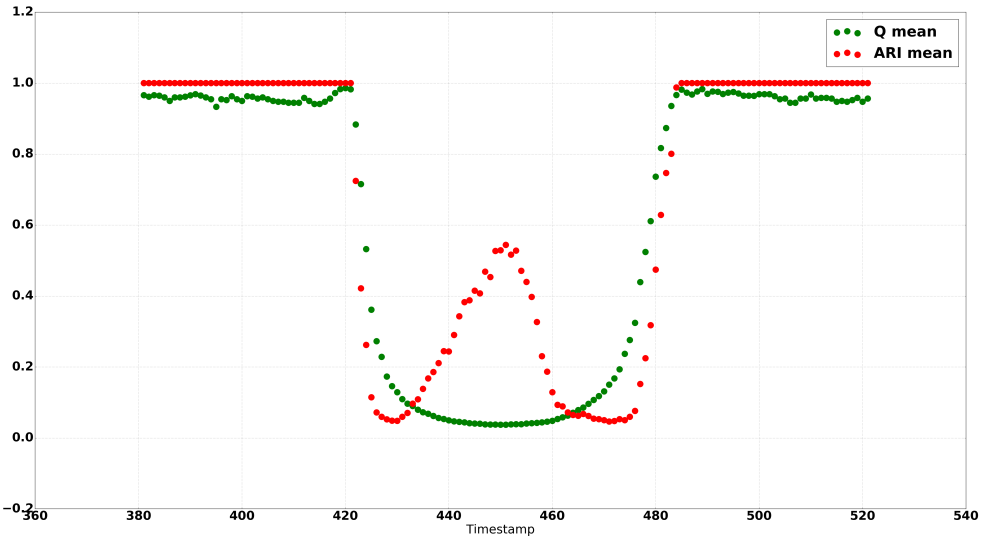


Figure 16: Variation of modularity and ARI along timestamps.

Another interesting relationship exists between Q, the ARI and the number of nodes per community, in-fact if the number of nodes per cluster is low the structure similarity will be trivially close to 1, as regard the modularity a lot of small but disconnected communities will also have an high modularity.

Figure 17 shows that as expected the pre and post activation have few nodes per community thus Q and ARI are high. Surprisingly in the core activation not only the ARI rises back but also the density (i.e. nodes per community) rises which means that **the communities are big and consistent in structure even though the modularity is low**. The density of communities shown is an average for all the 12 samples, the standard deviation is below 1.14 thus, given the strong consistency, in later analysis the nodes per community are not averaged over all the timestamps.

The reason of the relationship between a low Q and a high ARI has to be found in the distribution of correlation coefficients in a network and relies on the visual identification of the communities. In order to understand what is the relationship between a low Q and a high ARI, the static clustering obtained running Louvain in each window are visualized such that each node in a community has its own color. Humans have an extraordinary ability in recognizing patterns in images and this is why this type of analysis and the whole area of data visualization are of tremendous importance when dealing with big data and in order to interpret conflicting results.

In Figure 18 it can be seen that any time the ARI grows the communities became more and more defined and self contained in an area instead a low ARI is representative of a random

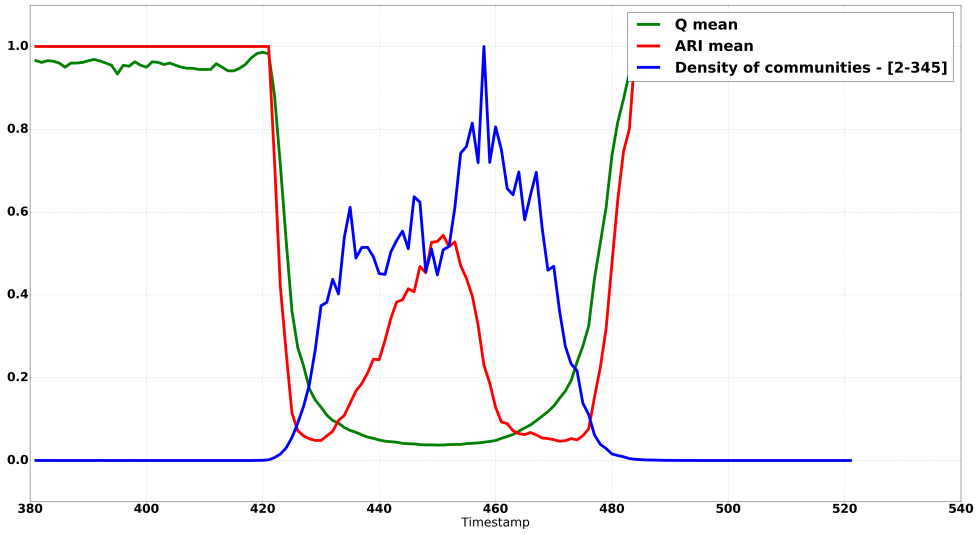


Figure 17: Variation of modularity, ARI and nodes per community over timestamps.

association of nodes to communities. In the core of the activation the modularity express the fact that the identified communities are not well separated among each other, despite this fact, the high ARI suggests that in each Louvain run the structure of the communities is very similar because nodes belong to the same cluster consistently. This analysis leads to the conclusion that the ARI is a good indicator of the quality of the clusters, that are, by a visual inspection, meaningful. Figure 19 shows the typical shape of the communities in a young mice and Figure 20 shows the typical shape of communities in old mice. The reason of the low modularity has to be attributed to the low standard deviation among the weights during the activation.

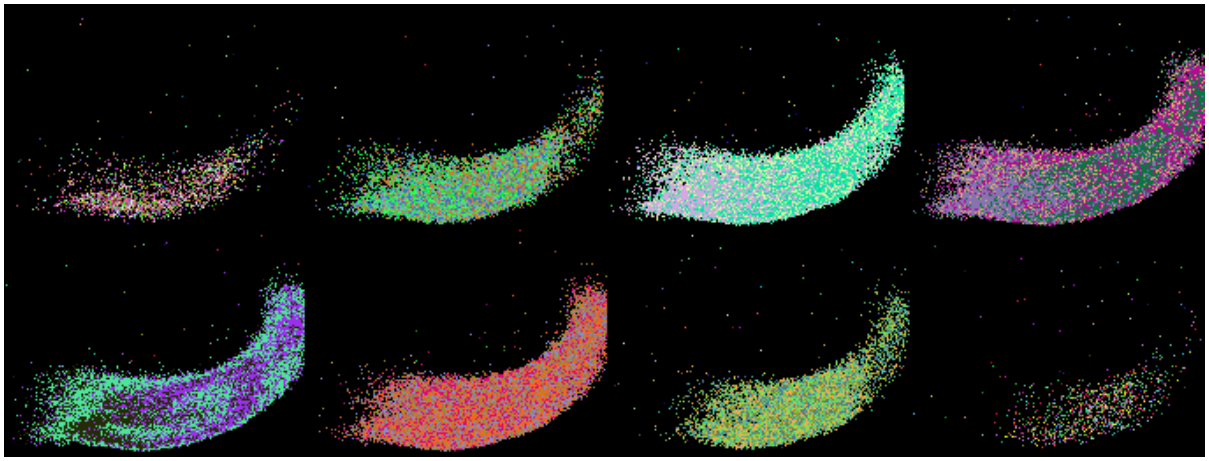


Figure 18: Louvain communities during time.

This analysis has a very important outcome as regards the meaningful timestamps of an activation. Until now all the result have been build on the execution of the pipeline (generation of correlation network, Louvain clustering) considering all the 1000 timestamps in an activation. The study of the Louvain communities highlights an interesting outcome: on average the number of meaningful timestamps (those in which brain communities are identified by Louvain) are 80. This result has been established by a double approach: first using the Q value, the ARI and the number of nodes per communities as shown in fig Figure 17 it is possible to identify the core activation timestamps, furthermore all core activations have been checked visually to establish if those timestamp match the ones in which Louvain could identify the communities.

With this method is possible to cut down the number of timestamps analyzed from 1000 to 80 which gives an enormous computational advantage. In order to select which timestamps to cut out, the number of nodes per timestamp has been adopted as an effective and conservative

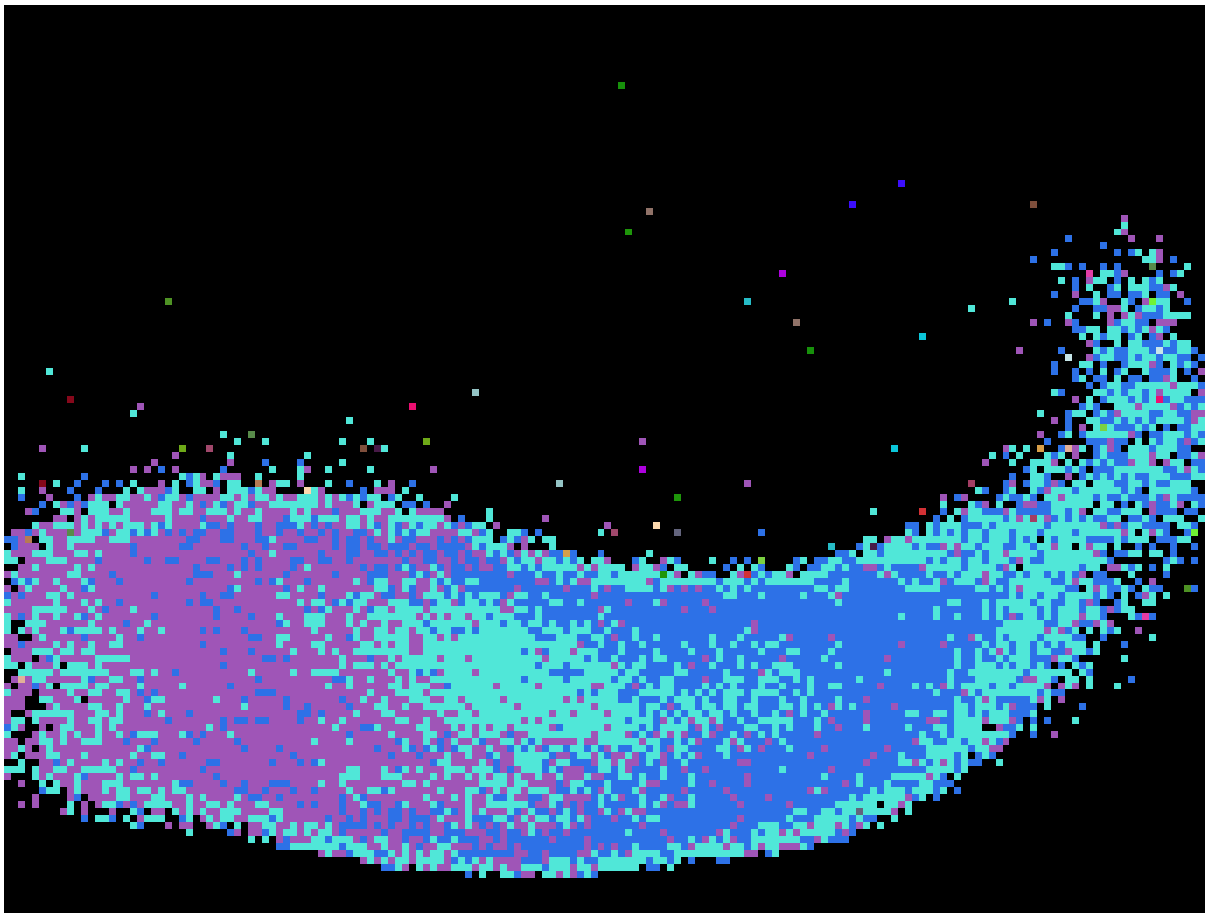


Figure 19: Communities identified by Louvain during a core activation of a young brain mouse.

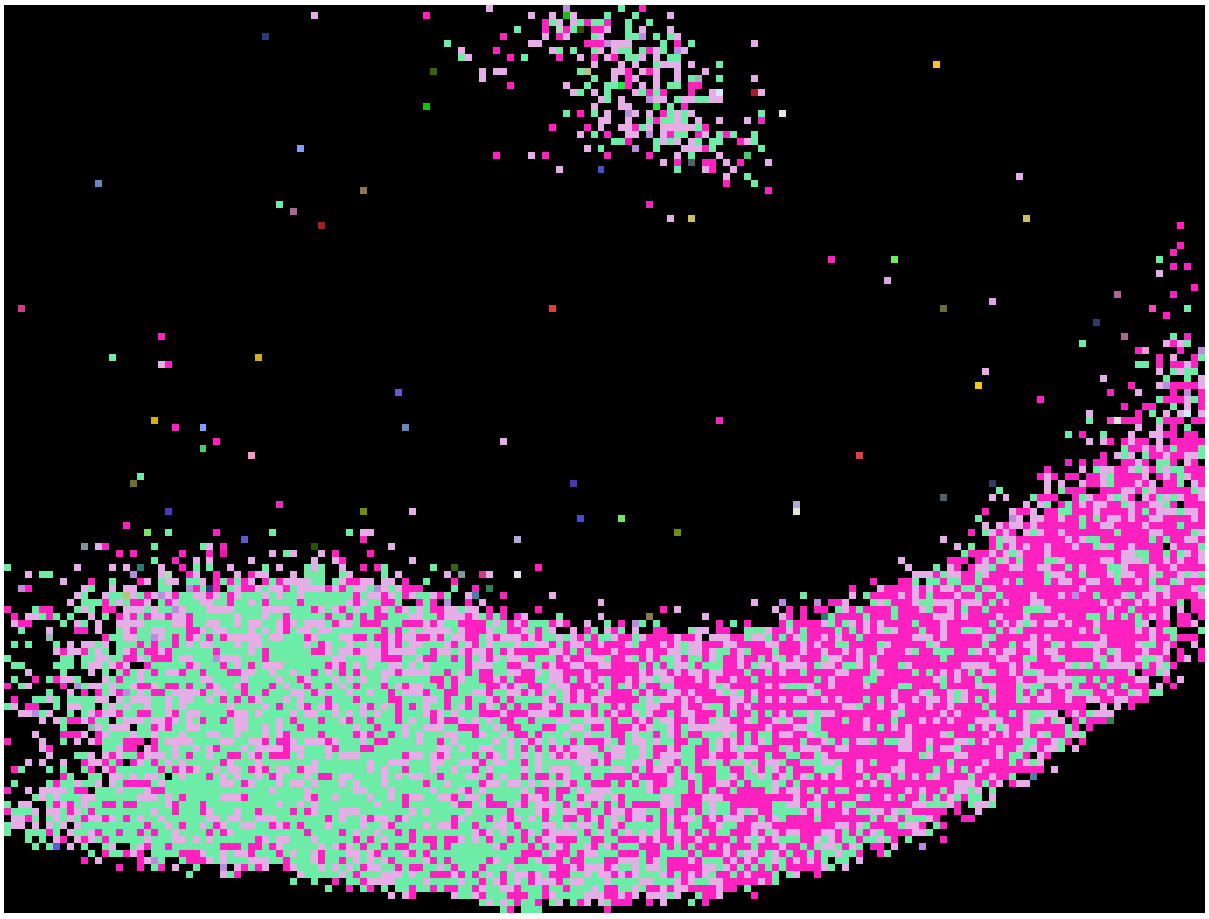


Figure 20: Communities identified by Louvain during a core activation of a young brain mouse.

measure, in fact it highly captures the beginning and ending of an activation in all the 80 activations of young and old mouse brains. An example of the number of nodes over time is shown in Figure 21.

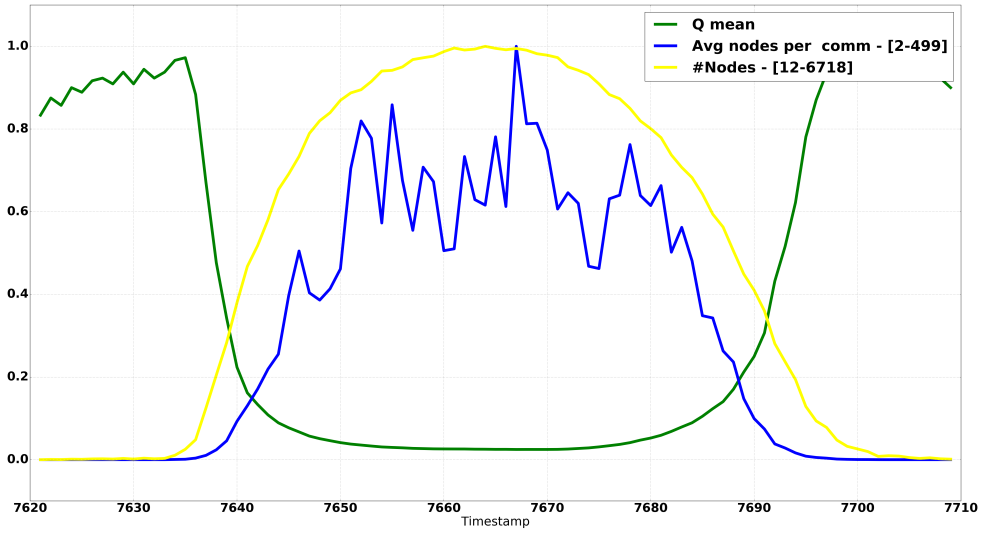


Figure 21: The normalized number of nodes per timestamp easily capture the presence of the core activation.

The number of nodes has been used to cut off every useless timestamps, thus highly improving the speed of the Commdy algorithm and statistics steps.

An even greater result is that there is no need to compute the correlation network and the number of nodes per timestamps but it is sufficient to look at the time-series of the pixels to

cut off the useless timestamps.

The result here is surprising and can shed lights on the actual meaning of the communities. One question that may arise from this study is: in which section of the pixel timeseries are the communities found? Is this section consistent among young and old brains?

Figure 22 shows the result of this investigation: the black line represents the average over all the pixels of their values along time (normalized and smoothed), the blue line represents the average number of nodes per community, the green line is the Q value, and the yellow line the number of nodes per timestamp. This picture is generated from the analysis of one brain but the graph is consistent for all the activations. One could expect that the communities would be found during the main activation, in this case between timestamps 7700 and 8000, instead the communities are found only in the very beginning of the activation. Notice also that because the networks are thresholded and are the result of a sliding window of 50 timestamps, this result only means that no network can be find with those characteristic in any other section of the timeseries. This can be interpreted as the fact that the information that is collected by our pipeline is based on the latency among the firing neurons.

5.0.5 Null model test

The previous sections proves that the Louvain algorithm is stable with respect to the modularity values although the value itself is too low to consider the static communities meaningful. The structural analysis indicates the ARI as a good index to understand if the communities are consistent, anyway the issue of a null model to test the significance of those communities is needed. In this section two null models are built in order to prove whether the communities

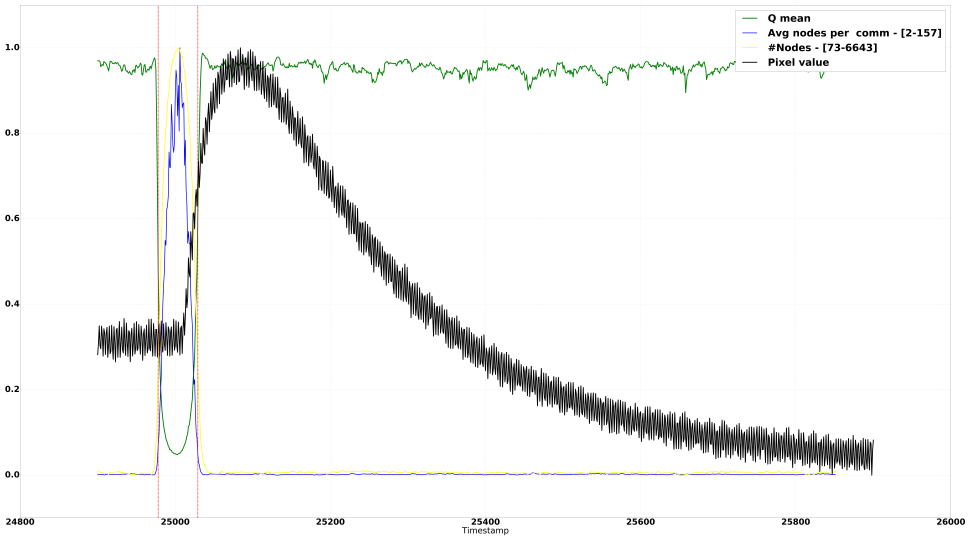


Figure 22: The normalized number of nodes per timestamp easly capture the presence of the core activation.

identified are 'special' or we could identify the same clusters' shapes with any other network.

The first model is a permutation model in which, given the real networks, the weights of the edges are shuffled, in this way we can test if the clustering depends on the particular value of edges, notice that this is a particularly strong model (baseline) because the topology of the network is exactly the same and the weights are implicitly taken from the same distribution. This model is then enhanced by analyzing the communities in the case the edge weights are all set equals to 1 and in the case the weights are randomly picked from a uniform distribution between 0 and 1.

The second model is a synthetic generated network in the form of a mesh in which each pair

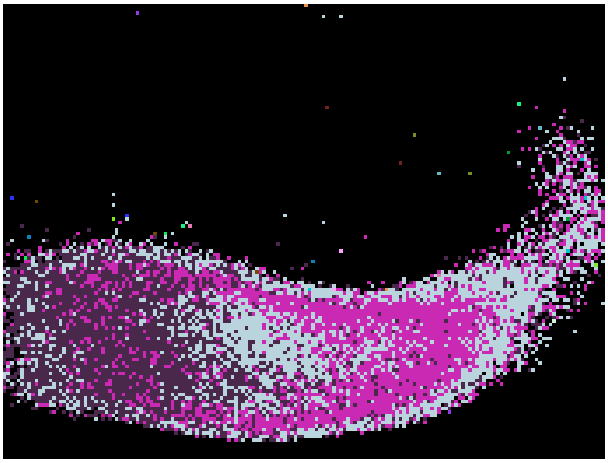
of nodes closer than some distance r are connected. This is used to test whether the clusters' shapes depend on this kind of meshed network structure and, if this is the case, it suggests hypothesis to identify problems in the generation of the network. From another point of view, there might be no problem about the data and the structure identified could highlight the underling wiring of the brain network.

The result of the shuffling null model generates the same clusters of the real network, the same outcome holds for the two variations on this model thus proving that **the shapes are totally independent from the network weights**. Figure 23 shows an example comparison of the cluster identified by the real network and the three null models.

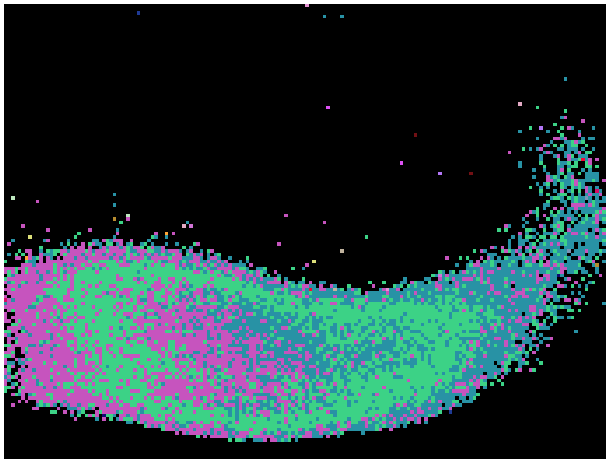
The synthetic mesh is generated building a 130x172 mesh and connecting every pair of nodes such that their euclidean distance is less than r, with $r \in [1..20, 30, 40, 50]$. Louvain is than used to identify communities in this mesh and the result is visually checked to see if the structure resemble the one of the Young or Old clusters. The cluster identified on the mesh network with r=1 are shown in Figure 24, and the communities grow in size (thus diminishing in number) as r increases. Empirically it has been found that with r=100 two communities are found which split the network exactly in two parts, moreover the Q value is 0.

The results obtained by those two null models clarify the nature of the brain communities:

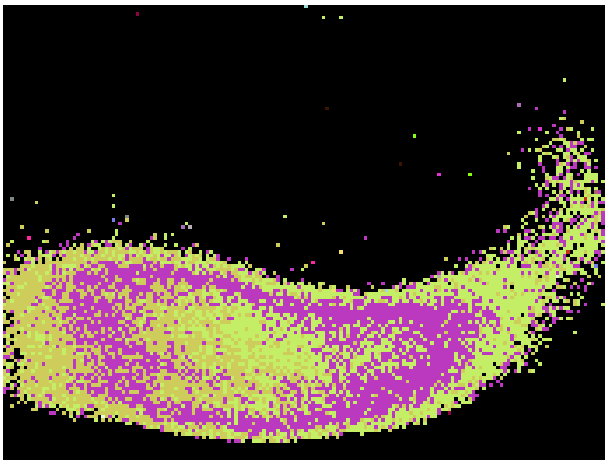
- The communities are independent of the edge weight distribution.



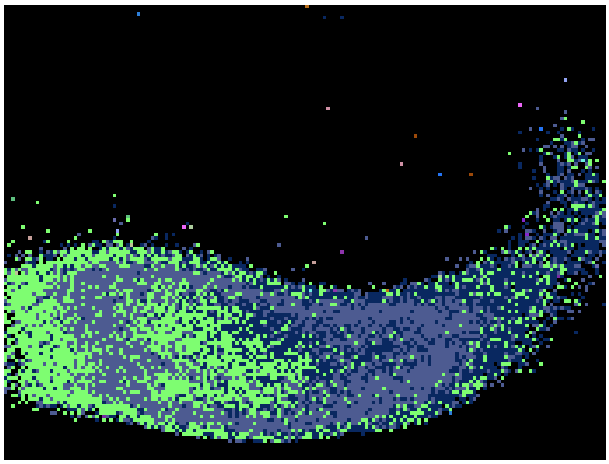
(a) Communities extracted with Louvain from the real network.



(b) Communities extracted with Louvain from the network with the shuffled edges.



(c) Communities extracted with Louvain from the network with edge weights fixed at 1.0.



(d) Communities extracted with Louvain from the network with randomly picked edge weights from a uniform distribution between 0 and 1.

Figure 23: Comparison of communities shapes given the same network structure and modifying the values of the edge weights. Colors do not have a meaning between figures but identify nodes belonging to the same community in a given picture.

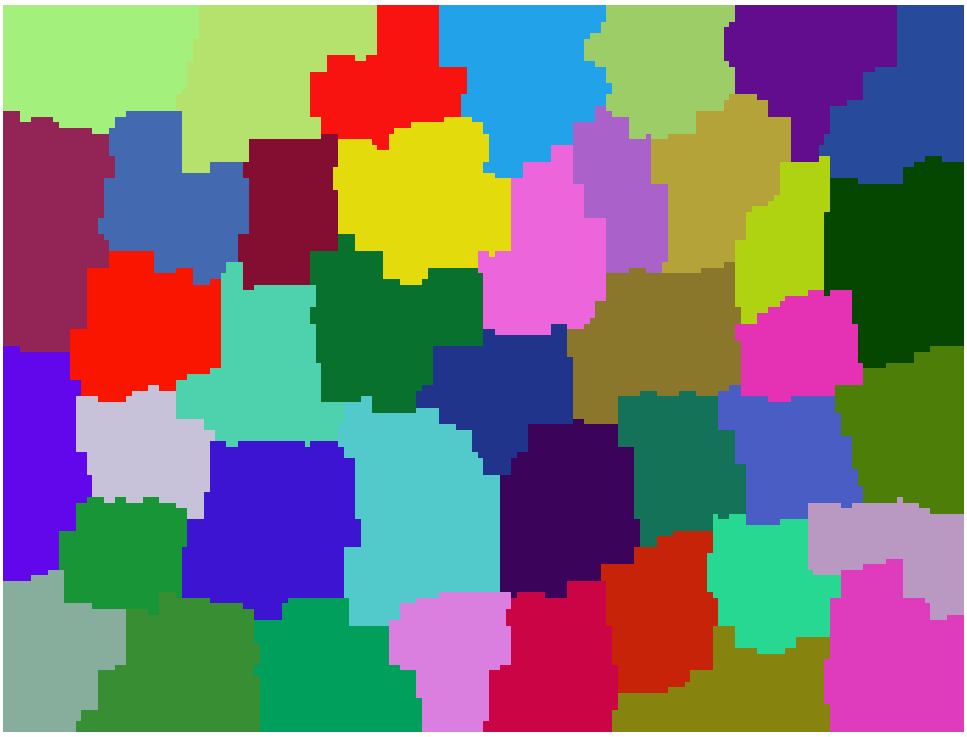


Figure 24: Community detected by Louvain on a mesh where node closer than r=1 are connected.

- The topology of the brain network is the key factor for the communities found by Louvain and the structure those not resembles a mesh but it's the outcome of a peculiar brain wiring.

5.1 Resolution Limit

The resolution limit problem arises when modularity optimization algorithms (such as Louvain) are used to find communities, in this cases it has been shown [50] that the size of clusters found depend on the size of the network itself, thus those algorithms won't detect communities of highly different sizes even though they might be inherent in the topology of the network. Louvain is subject to the resolution limit problem and in order to verify if this could interfere with the identification of the community another community detection method as been adopted. It is important to notice that by comparing Louvain with Infomap, not only the resolution problem is faced and quantified but a complete new approach (based on flow) is used to identify communities thus validating again the robustness of Louvain.

5.2 Comparison of Louvain and Infomap

5.2.1 Infomap Description

Infomap is a community detection algorithm that optimizes the map equation, first introduced in [47].

This algorithm tries to capture the flow in the network thus is useful when trying to detect communities in weighted and directed networks which represents constraint on the flow. Anyway this algorithm can be used also for unweighted and undirected graphs.

The algorithm is robust with respect to the resolution limit problem in its two-level formulation because the limit "depends on the total weight of links between communities rather than on the total weight of all links.[...] Moreover, for many networks the resolution limit vanishes completely in the multilevel formulation of the map equation"[48].

The algorithm leverages the idea of random walks on a network and identifies communities by solving a coding problem. In this analysis the algorithm has been run using default parameters and specifying weighted undirected links. The input of the algorithm is the correlation network (link list format), the output is a (.map) file which describes the best two-level partition (no hierarchical structure is preserved). It is worth to notice that the core of the algorithm follows the Louvain method but it optimizes the map equation instead of the modularity, this reflects the different approach in the two community detection algorithm: Louvain empathize topology, Infomap network flow.

Infomap algorithm is based on the map equation which exploits code theory in order to obtain the community structure on a network, be W = module partition of k nodes $\in [1, 2, ..., k]$ into c modules $i \in [1, 2, ..., c]$,than the code length lower bound is L(W).

$$L(W) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{i=1}^{c} p^i_{\circlearrowright} H(\mathcal{P}^i)$$

where $q_{\curvearrowright}$ is the probability that the random walk switches modules on any given step. $H(\mathcal{Q})$ is the entropy of the module names.$H(\mathcal{P}^i)$ is the entropy of the within-module movements. The weight $p^i_{\circlearrowright}$ is the fraction of within-module movements that occur in module i, plus the probability of exiting module i.

The equation is composed by two component: the first represents the entropy of the movement between modules, whilst the latter is the entropy of movements within modules. Each is weighted by the frequency with which it occurs in the particular partitioning.

5.2.2 Communities identified by Infomap

The Infomap algorithm has been run on 7 young mice activations and 7 old mice activations, the results obtained reveal that Infomap cannot identify any community in the brains but it does only identify the all auditory cortex and sometimes the thalamus as one huge community. Figure 25 shows an example of communities identified by Infomap during time, anyway the poor performances of Infomap on this dataset could be foretasted by considering that the communities found by Louvain were a direct result of the structure and topology of the network and were completely independent on the weights direction and flow. Given that Infomap is based

on the concept of flow in a network the results on this dataset are meaningless.

Given that the results of the static community analysis were useless no attempt as be made to run CommDy on the Infomap communities.

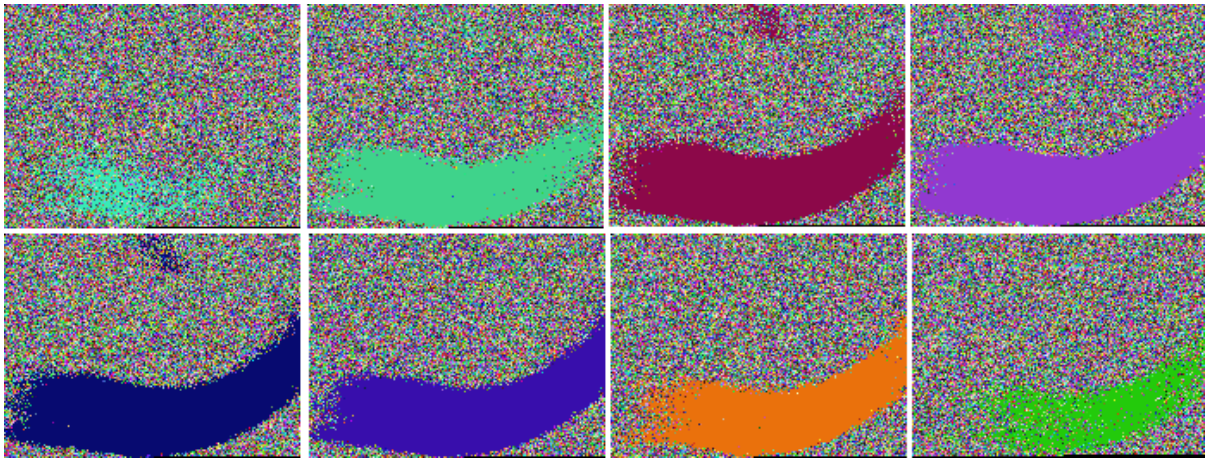


Figure 25: Communities identified by Infomap during a core activation of a young brain mouse.

CHAPTER 6

COMMDY ROBUSTNESS ANALYSIS

6.1 Sensitivity to Louvain runs

In the previous chapter it has been shown that the Louvain Algorithm is stable and the communities obtained are meaningful compared to the ones extracted by the Infomap algorithm.

Nonetheless because there are variations in the communities identified by Louvain during different runs we want to asses the impact of this variability on the results produced by Commdy. The question that we want to address is whether the dynamic communities identified by Commdy for a given node are consistent among the Louvain runs or if a slight variation of Louvain output creates a huge difference in the dynamic communities. In order to do so, the Adjusted Rand Index (ARI) is used to evaluate the overlap in the dynamic communities for a single node, in other words for each node two communities generated in two runs are compared. As explained in Chapter 3, a run of Commdy based on the result of a Louvain run produces the dynamic communities. For each node, Commdy identifies the community to which it belongs in each timestamp.

In order to asses the sensitivity of Commdy with respect to the Louvain outputs, the Commdy algorithm has been run on 15 different runs of Louvain for each of the 12 samples: 6 young activations and 6 old activation. The robustness matrix has 15 rows and 15 cols which represents

15 runs of Louvain. A cell (i, j) in this matrix is the average of the ARI among all the nodes ever considered by Louvain. Formally:

$$cell(i,j) = \frac{\sum_{n \in N} ARI(comm(i,n), comm(j,n))}{|N|}$$

where N is the set of nodes considered at least once by Louvain, $ARI(x, y)$ is the Adjusted Rand Index among x and y, $comm(i, n)$ is the dynamic communities of the node n in the run i. The mean ARI among two runs is the average over nodes of their ARI, thus a robustness matrix is created (fig. Figure 26). The matrix already shows the overall high similarity of the communities identified from different Louvain runs infact the average ARI (on the matrix) is 0.715 with a standard deviation of 0.074.

By itself the result obtained comparing the runs seems valuable, anyway in order to asses the real significance of the result three random models are created. The three models were build each one answering a precise question:

1. What is the ARI if the partitions were randomly created with the same number of timestamps and the same number of communities (Commdy colors)?
2. What is the ARI if the partitions were randomly created with the same number of communities but half of the original timestamps?
3. What is the ARI for random partitions created exactly from the same data, by shuffling the communities?

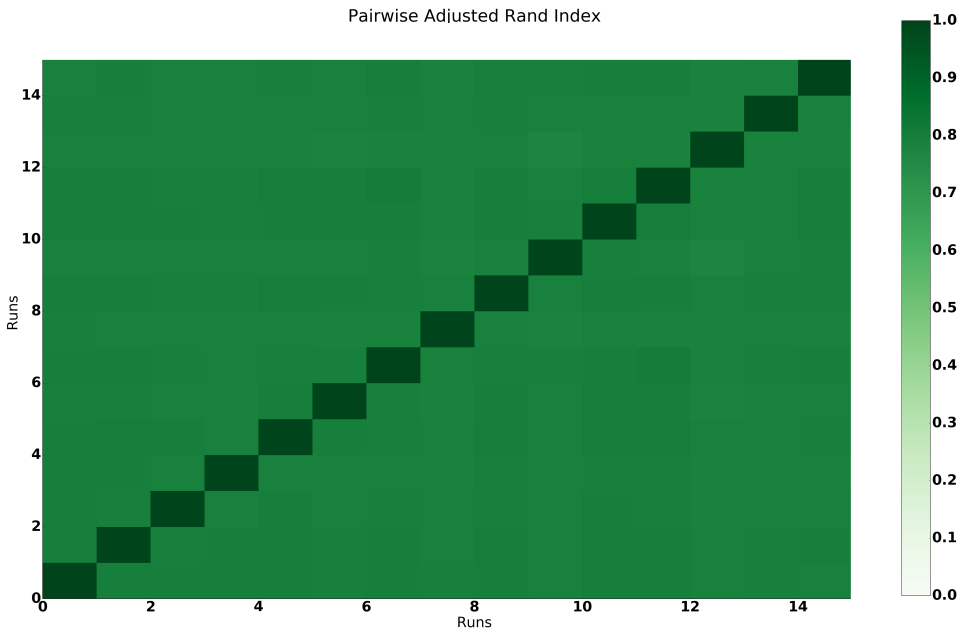


Figure 26: Robustness Matrix of Commdy on several Louvain outputs. Each cell represents the average ARI among nodes of two runs. Overall the average ARI is 0.715 with a standard deviation of 0.074, which is a high indicator of the robustness of Commdy over several Louvain runs.

The results are reported in Table I. Each row represents a sample for which is indicated the category of the mouse (Young/Old), the mean and standard deviation of the ARI, the mean ARI of the null model 1 and 2. The results for the null model 3 are not reported since the ARI was trivially 0 in each case.

There are several point to mention: first of all the standard deviation is reported because the

TABLE I: MEAN AND STANDARD DEVIATION OF THE ARI BETWEEN DYNAMIC COMMUNITIES. RESULTS REPORTED FOR 6 YOUNG AND 6 OLD MICE BRAINS. EACH ROW REPRESENTS A SAMPLE FOR WHICH IS INDICATED THE CATEGORY OF THE MOUSE (YOUNG/OLD), THE MEAN AND THE STANDARD DEVIATION OF THE ARI, THE MEAN ARI OF THE NULL MODEL 1 AND 2.

Type	**Mean ARI**	**STD ARI**	**Null Model 1**	**Null Model 2**
O	0.81	0.05	0.06	0.71
O	0.80	0.05	0.06	0.65
O	0.74	0.07	0.06	0.67
O	0.62	0.10	0.08	0.86
O	0.66	0.09	0.07	0.81
O	0.69	0.08	0.07	0.83
Y	0.76	0.06	0.06	0.62
Y	0.73	0.07	0.07	0.63
Y	0.68	0.08	0.07	0.77
Y	0.72	0.07	0.07	0.79
Y	0.71	0.08	0.07	0.74
Y	0.66	0.09	0.07	0.78

mean ARI is created by averaging the ARIs in the 15 x 15 matrix (in which each cell is the ARI among two runs), anyway the standard deviation is very low for each row, thus it is a first

indicator of the robustness of the results. The mean of the ARI is always above 0.62 which by itself is a significant ARI, moreover the importance of this result is strengthen by the results of the null models. The first one, which represents a null model created using random partitions with the same number of communities and timestamps, is extremely low, infact the ARI has been originally chosen for the ability to account for chance (random partitions). More importantly when we half the number of timestamps used to randomly create partitions (null model 2) the value of the ARI is very high and often higher than the real results. This means that as we increase the number of timestamps the probability of the ARI of being high (by chance) decrease significantly, thus the result obtained for our model is highly significant. **With this result it is possible conclude that different runs of Louvain do not create significant difference in the dynamic communities identified by Commdy which cannot be the result of chance**.

6.2 Sensitivity to the costs

The Commdy algorithm is based on several assumptions (see 3.1.4) which directly translate into the 3 costs that are given as arguments to the algorithm: switching, absence, visiting. Because there is no study on the parameter space neither neuro-scientific principle to directly set those costs, the parameter space has to be investigated.

Given the analytical study of Commdy carried on in [7] it is known that, the important measure, which influences the community structure, is the ratio among the switching and vising cost, for this reason the space of parameter is analyzed for the costs shown in Table II.

TABLE II: COST SPACE EXPLORED FOR COMMDY.

Switching	**Absence**	**Visiting**
1	1	1
1	1	3
1	3	1
3	3	1
1	3	3
3	1	3
5	1	1
1	5	1
1	1	5

The sensitivity of Commdy with respect to the costs is assessed in the following directions:

1. The difference among the communities generated with different costs is quantified calculating the ARI between them.

2. Principal Component Analysis (PCA) is used to determine if there is a clear difference between old and young brains, and identify meaningful components.

3. A random forest classifier is built to distinguish between old and young brains: to classify brains based on the Young - Old dataset.

4. A biological hypothesis is formulated to explain the differences between young and old brains, and a set of classifiers are trained on a different dataset (which is the result of applying increasing level of drugs on the brains) and tested on non-drugged brains, to verity the hypothesis. The accuracy is evaluated for each cost.

6.2.1 Robustness of the dynamic communities

The dataset used to calculate the ARI is composed of 62 activation: 24 old and 38 young. For each activation Commdy is run with the 9 different set of costs, the dynamic communities generated by each cost set are finally compared using the ARI. The mean and standard deviation for each run of Commdy is reported in Table III.

This result highlights an important consideration: the average ARI for the old brains is 0.52 whilst for the young 0.62, this results strengthen the finding that clear community structures can be found only in young brains (as shown from the Louvain algorithm) and those community are much more consistent that the one found for old brains which consequently have a lower ARI.

Once again we can increase the confidence in this results by exploiting visualization, in this case by plotting the dynamic communities identified by CommDy. Figure 27 shows the typical expression of dynamic communities (same colors represents the same community in different windows) for old brains whilst the ones identified for young ones are in Figure 28. One can notice that old brains hardly synchronize to form a stable community and this process is much slower than young brains, on the contrary young brains synchronize fast to form a stable and continuous community. Thanks to this visualization technique it is possible to understand why young dynamic communities are much structurally coherent (higher ARI) and that despite the 0.52 of mean ARI for the old brains, that is mostly due to the fact that old brains forms two communities (red and green) in the picture, there is not a clear division into communities.

TABLE III: MEAN AND STANDARD DEVIATION OF THE ARI AMONG DYNAMIC COMMUNITIES OVER DIFFERENT COMMDY COSTS.

Type	**Mean ARI**	**STD ARI**
O	0.61	0.23
O	0.64	0.20
O	0.62	0.22
O	0.65	0.19
O	0.52	0.30
O	0.62	0.21
O	0.51	0.29
O	0.48	0.32
O	0.48	0.32
O	0.48	0.31
O	0.48	0.32
O	0.48	0.32
O	0.49	0.32
O	0.47	0.32
O	0.54	0.27
O	0.50	0.29
O	0.50	0.32
O	0.50	0.29
O	0.52	0.28
O	0.47	0.30
O	0.49	0.32
O	0.48	0.32
O	0.47	0.33
O	0.52	0.21

Type	**Mean ARI**	**STD ARI**
Y	0.50	0.29
Y	0.63	0.21
Y	0.64	0.21
Y	0.69	0.18
Y	0.56	0.25
Y	0.66	0.19
Y	0.66	0.19
Y	0.62	0.21
Y	0.66	0.19
Y	0.62	0.22
Y	0.47	0.33
Y	0.49	0.30
Y	0.54	0.26
Y	0.59	0.24
Y	0.59	0.24
Y	0.62	0.23
Y	0.72	0.16
Y	0.71	0.17
Y	0.76	0.14
Y	0.49	0.30
Y	0.52	0.28
Y	0.52	0.28
Y	0.48	0.30
Y	0.51	0.29
Y	0.49	0.27
Y	0.55	0.26
Y	0.54	0.27
Y	0.54	0.24
Y	0.56	0.26
Y	0.53	0.26
Y	0.48	0.30
Y	0.54	0.26
Y	0.56	0.24
Y	0.69	0.18
Y	0.71	0.16
Y	0.70	0.16
Y	0.67	0.19
Y	0.66	0.19

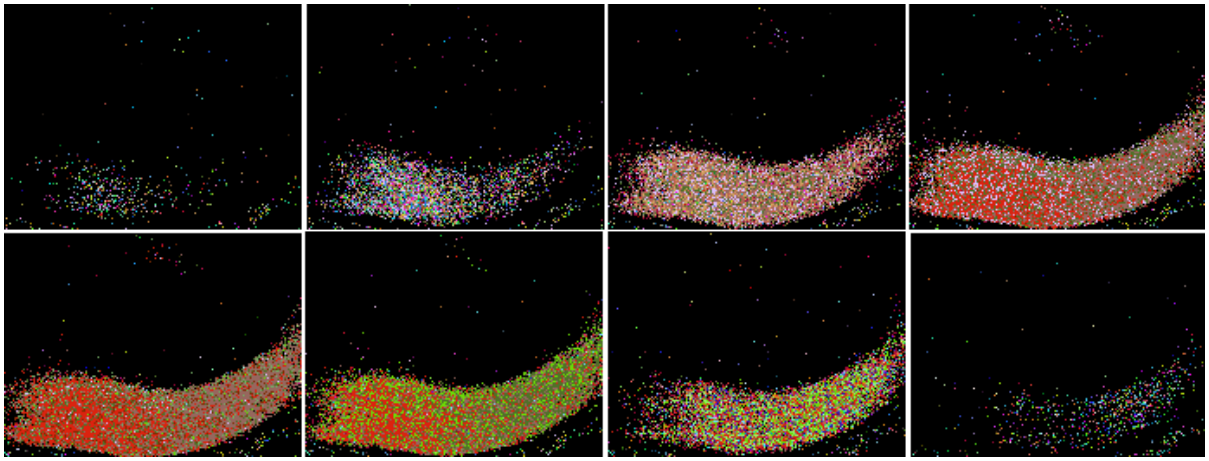


Figure 27: Dynamic Communities in old brains.

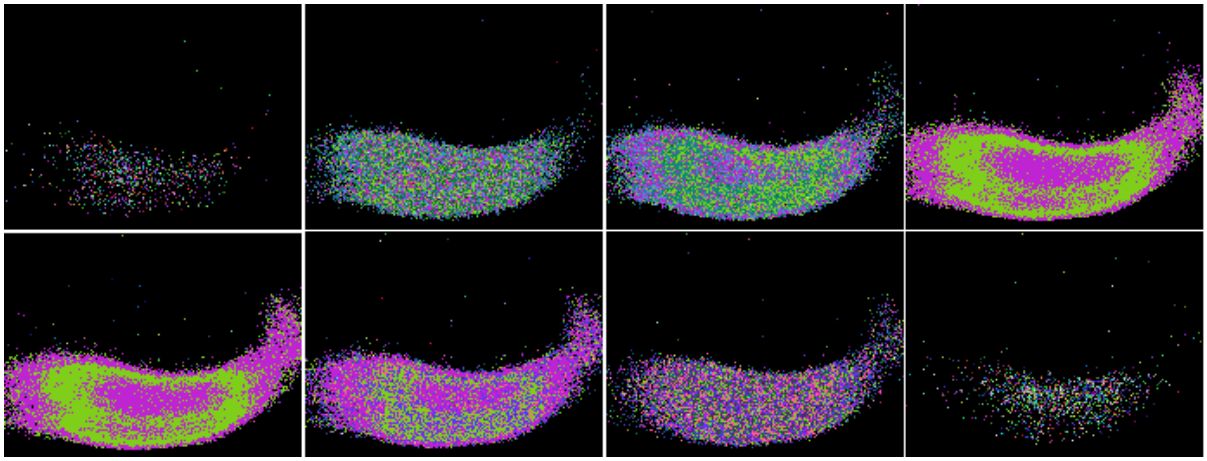


Figure 28: Dynamic Communities in young brains.

CHAPTER 7

BRAIN CLASSIFICATION

7.0.1 Principal Components Analysis (PCA)

The same dataset used in the previous section is used to analyze the data through the PCA method. Each of the CommDy dynamic communities generated in the previous step is used to extrapolate measures as explained in Section 3.1.5, this allows for the quantification of the differences among the two categories and to spot differences that can highlight insights for the neuroscience community.

For each set of cost the PCA algorithm is run on the statics collected from the brains and the results are shown in the plots below, where each dot represents a brain. Interestingly enough the two principal component explain around 90% of the variance together.

For each on this plots it is hard to tell which is the main variable to distinguish old and young brains, anyway what is evident is that, regardless the set of cost used, there is a clear separation between the two categories, this is again a confirmation of the fact that there is a difference between young and old brains, as already found analyzing the ARI and the CommDy visualization, and naturally introduce the next phase of the work: automatic classification.

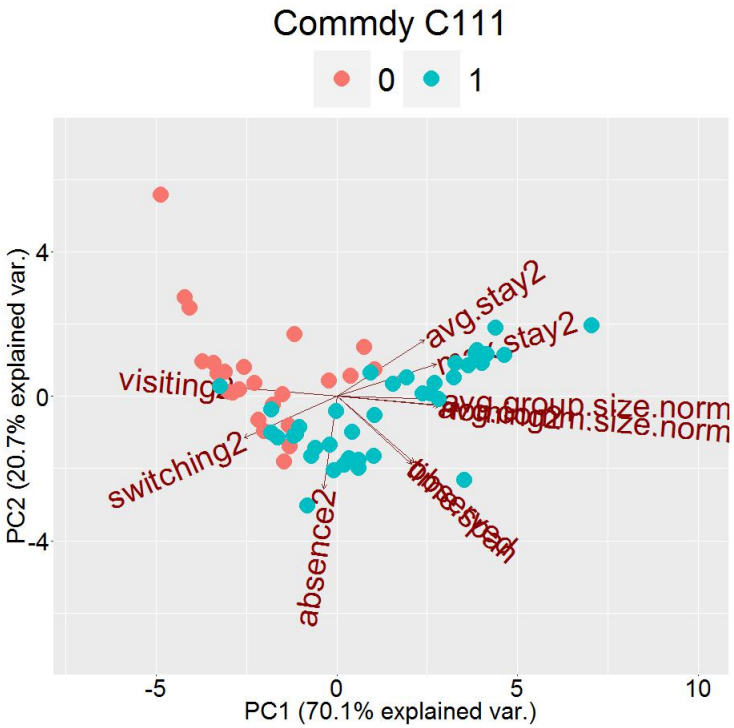


Figure 29: PCA Biplot, each point is a brain activation. The statistics are collected on the output of Commdy with costs 111.

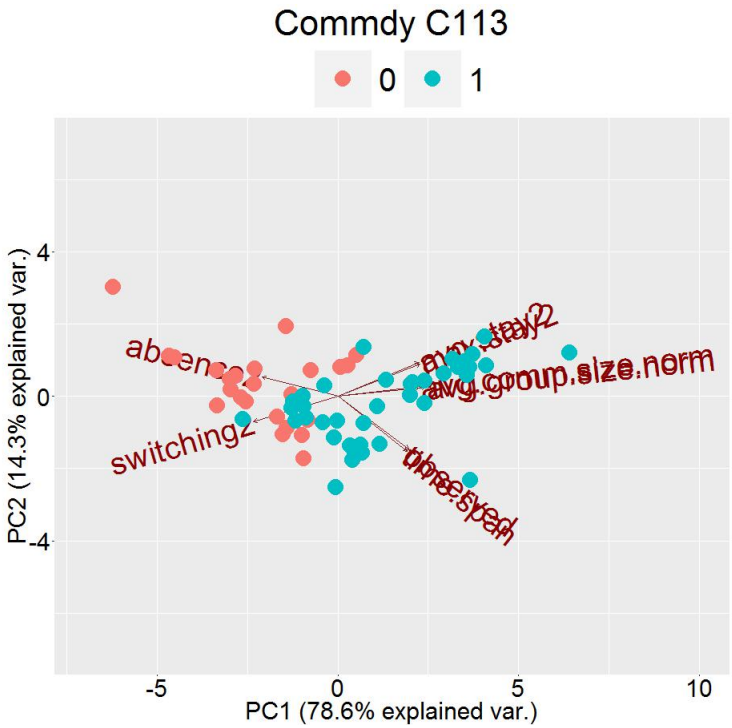


Figure 30: PCA Biplot, each point is a brain activation. The statistics are collected on the output of Commdy with costs 113.

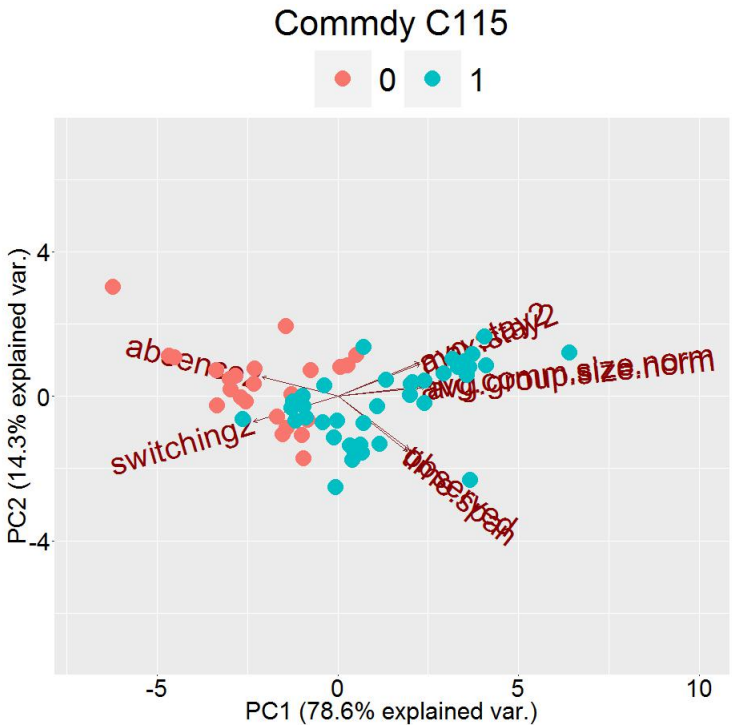


Figure 31: PCA Biplot, each point is a brain activation. The statistics are collected on the output of Commdy with costs 115.

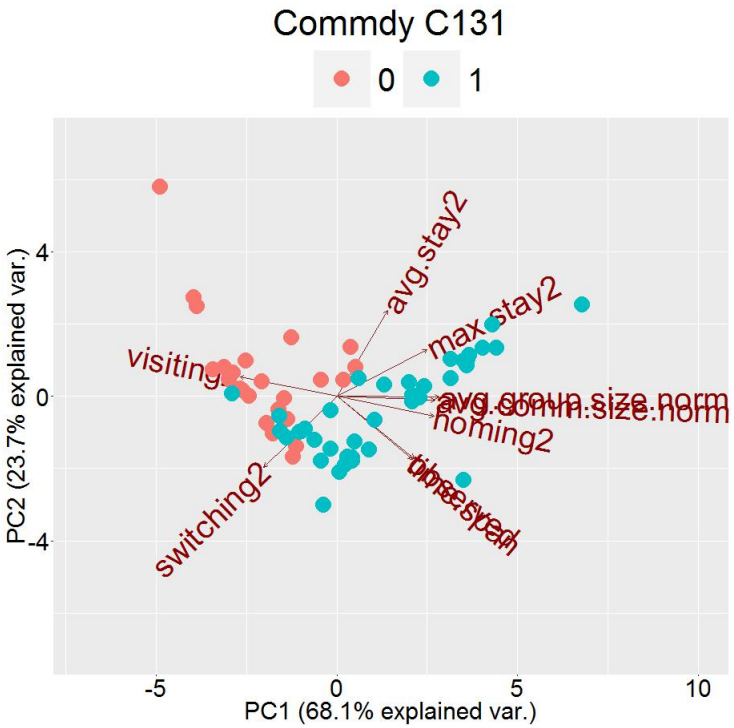


Figure 32: PCA Biplot, each point is a brain activation. The statistics are collected on the output of Commdy with costs 131.

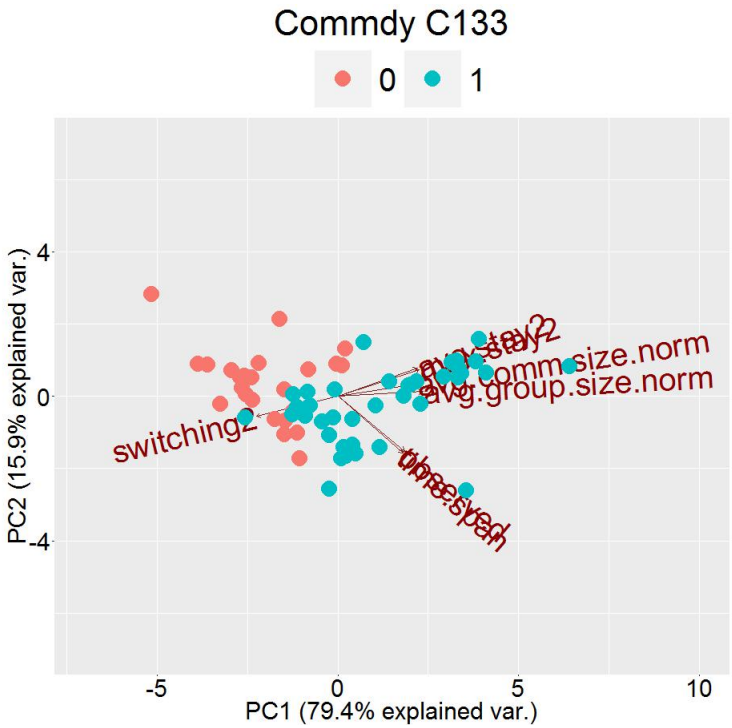


Figure 33: PCA Biplot, each point is a brain activation. The statistics are collected on the output of Commdy with costs 133.

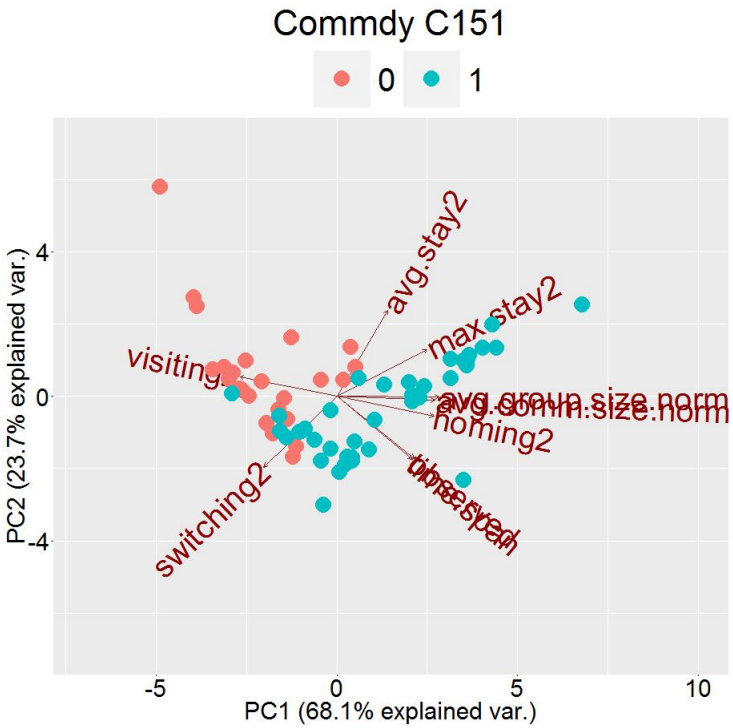


Figure 34: PCA Biplot, each point is a brain activation. The statistics are collected on the output of Commdy with costs 151.

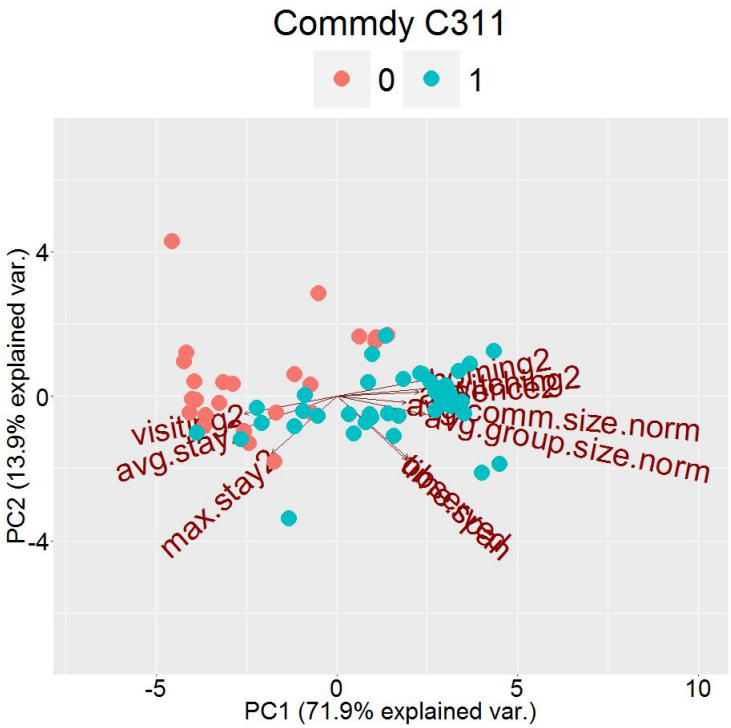


Figure 35: PCA Biplot, each point is a brain activation. The statistics are collected on the output of Commdy with costs 311.

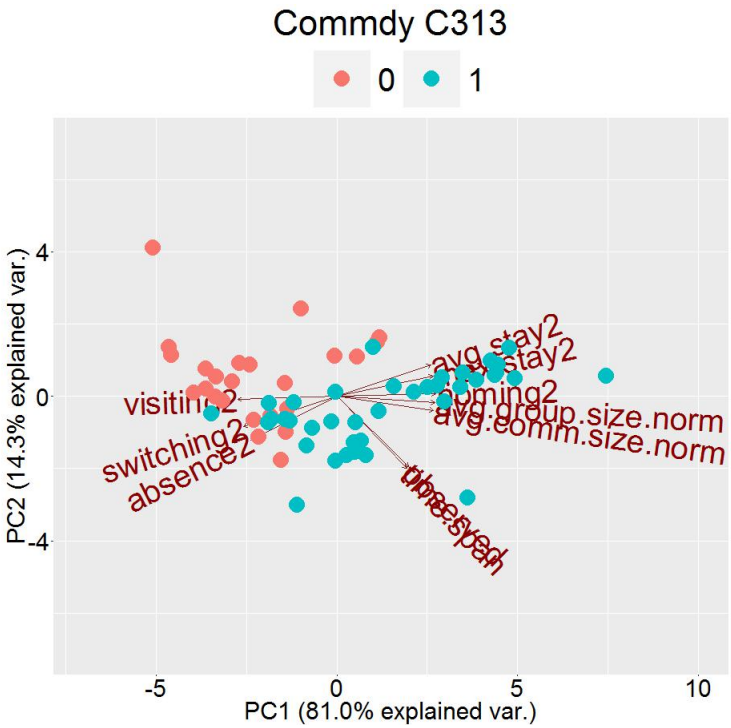


Figure 36: PCA Biplot, each point is a brain activation. The statistics are collected on the output of Commdy with costs 313.

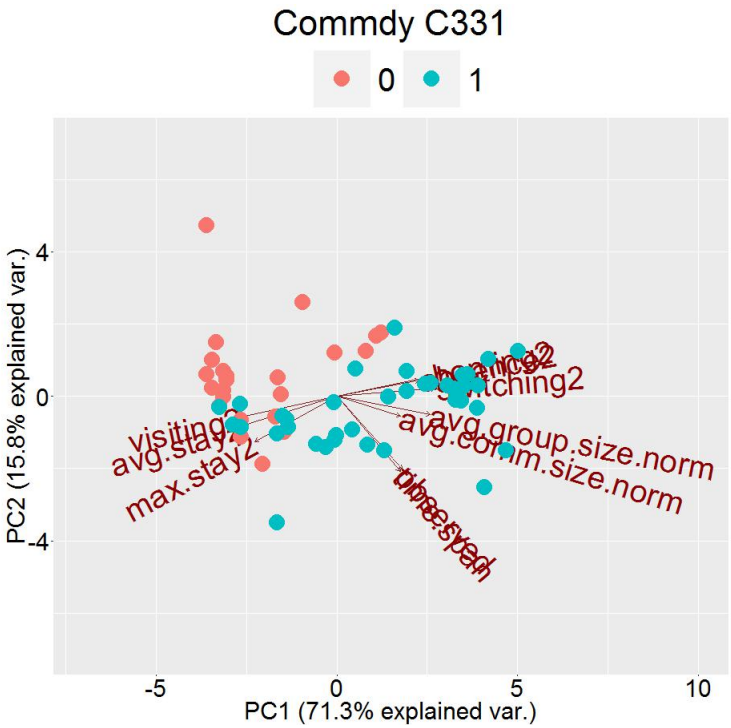


Figure 37: PCA Biplot, each point is a brain activation. The statistics are collected on the output of Commdy with costs 331.

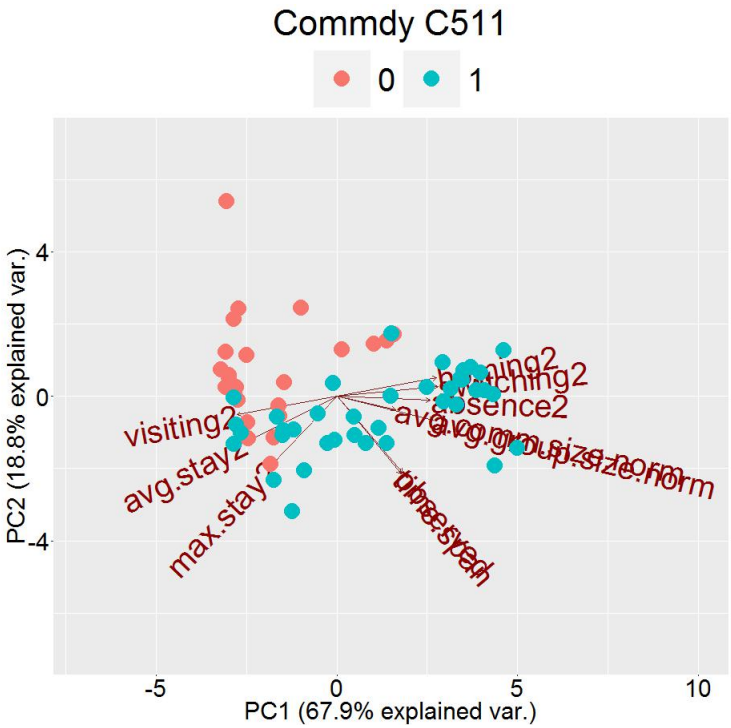


Figure 38: PCA Biplot, each point is a brain activation. The statistics are collected on the output of Commdy with costs 511.

7.0.2 Random Forest Classifier

PCA reveals the existence of distinguishable characteristics, thus in this section a Random Forest algorithm is used to classify the data, the machine learning algorithm is chosen because of the transparency of the algorithm which allows for the interpretation of the results.

The first classifier is based again on 24 old brains and 28 young brains. The Random Forest is trained using 50 trees, in order to be able to model the data without overfitting, on 12 parameters (Table V):

Group size, Span (duration), Community size, Switchiness, Absenteeism, Visiting, Average stay, Maximum stay, Community stay, Observed, Group size normalized, Community size normalized.

The Random Forest algorithm is inherently "randomic" for this reason each of the results that will follow have been obtained by running the training - test phases 20 times, moreover given the size of the dataset the leave-one-out cross validation has been chosen to give more reliability to the results.

For each cost set the random forest is trained using all the 12 parameters and using one parameter at a time in order to evaluate the importance of the single variable.

The accuracy of the algorithm is displayed in Figure 39 (standard deviation is < 0.01 for all cost sets).

The result show how for any set of cost the performances compared to the baseline (majority class) are significantly better. Moreover it is noticeable that any time the visiting cost is higher

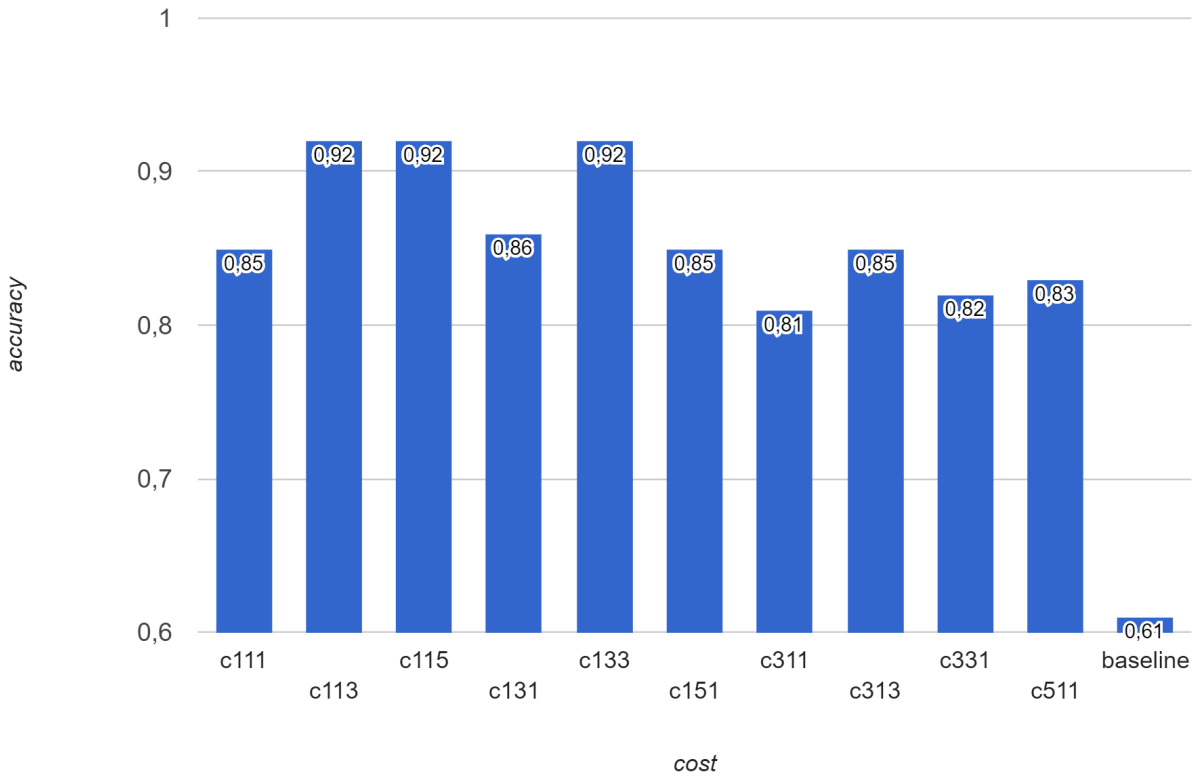


Figure 39: Random forest with leave one out cross validation.

than the switching cost the accuracy is higher, this is sound because when the visiting cost is higher nodes 'pay' a higher price to visit than to switch, in other words they prefer to stay with their community or to totally change community, this effect creates communities that are much more 'solid' thus leading to better performances for the classifier.

By training the classifier on one variable at a time it is possible to estimate the importance of a variable in the classification task. Table IV shows the accuracy of each variable depending on the cost set, in bold the variable that obtains the higher accuracy is highlighted.

TABLE IV: IMPORTANCE OF VARIABLES FOR THE CLASSIFICATION TASK

Costs (S,A,V)	111	113	115	131	133	151	311	313	331	511
Observed	0.71	0.71	0.71	0.71	0.72	0.71	0.71	0.71	0.71	0.71
Span	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.74
Switchness	0.65	0.70	0.69	0.56	0.69	0.56	0.77	0.70	0.69	0.74
Absence	0.57	0.70	0.71	0.61	0.61	0.61	0.68	0.54	0.68	0.69
Visiting	0.74	0.61	0.61	0.69	0.61	0.69	0.75	0.71	0.72	0.62
Homing	0.74	0.61	0.61	0.69	0.61	0.70	0.74	0.71	0.72	0.62
Average group size	0.79	0.79	0.79	**0.79**	0.79	**0.79**	**0.79**	0.79	**0.79**	**0.79**
Average community size	0.64	0.82	**0.82**	0.77	0.81	0.77	0.77	0.75	0.64	0.5
Average stay	0.5	0.66	0.66	0.60	0.66	0.59	0.65	0.63	0.61	0.68
Max stay	0.56	0.71	0.70	0.45	0.71	0.45	0.42	0.71	0.66	0.52
Average group size normalized	0.79	0.79	0.79	0.79	0.79	**0.79**	**0.79**	0.79	**0.79**	**0.79**
Average community size normalized	**0.84**	**0.83**	**0.82**	**0.79**	**0.82**	**0.79**	0.75	**0.82**	0.56	0.54
All variables	0.85	0.92	0.92	0.86	0.92	0.86	0.81	0.85	0.83	0.83

It is possible to notice that average group size and the average community size are the most important accordingly to this analysis. The difference between group and community can be easily understood referencing Figure 2, and zooming on the steps 2 and 3 as shown in Figure 40. In particular the Average Group Size (AGS) is a measure of the dimension of the static communities found over the period of time analyzed, conversely the Average Community size (ACS) is a measure of the dimension of the dynamic communities, their normalized version is obtained just by dividing by the number of observed nodes. Formally, let's consider N the total number of observed nodes, G the total number of groups, W_i the number of windows in which community i is identified, X_i the total number of nodes in community i over time, C the number of communities, the following quantities are defined:

$$AGS = \frac{N}{G}$$

$$ACS = \frac{\sum_i \frac{X_i}{W_i}}{C}$$

The case is Figure 40 is a particular case in which AGS = ACS because $W_i \times C = G$ which is not the standard case.

The result that the ACS and AGS are a meaningful indicator for the classification task can give important insights to guide the research in the field of neuroscience. In particular it is possible to hypothesize, leveraging on these results and on the images produces by Commdy, that young mouse brains are different from old ones because the communities found in young brains are bigger whilst for the old brains the communities are smaller thus higher in number. There is a

lack of abilities for old brains to form stable and large communities which may be attributed to the slower synchronization and communication among neurons, in the next paragraph a test is presented in order to verify this hypothesis.

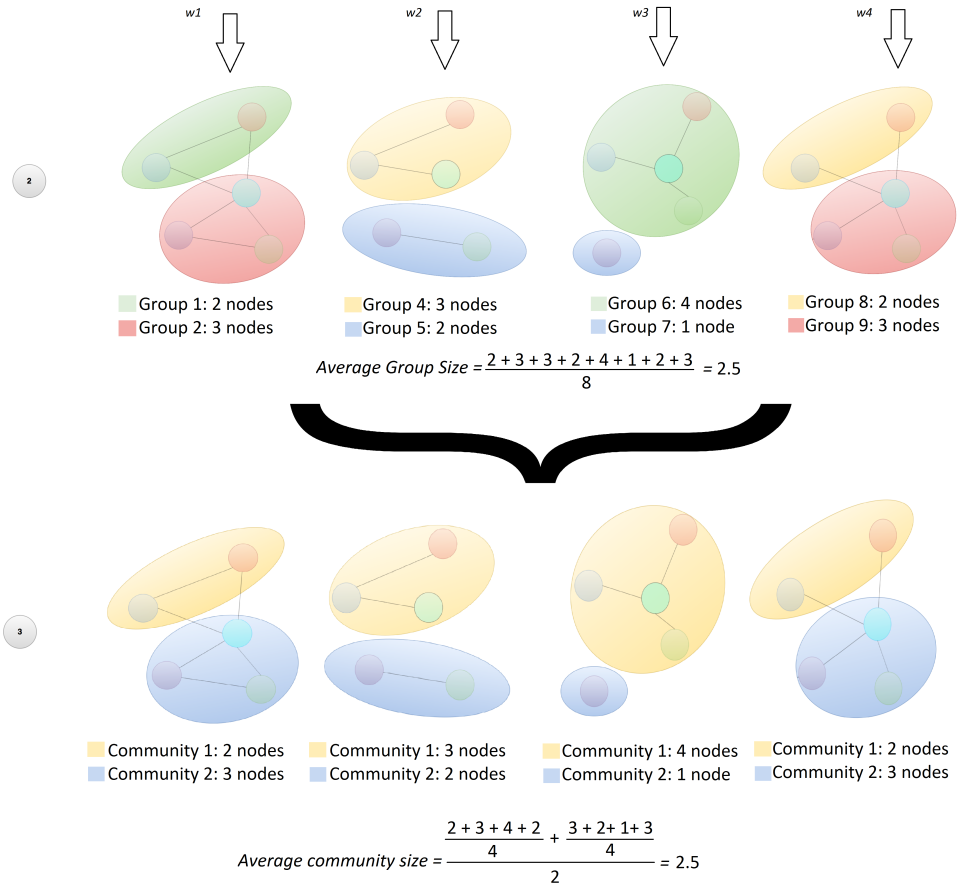


Figure 40: Group size average and Community size average example.

TABLE V: SAMPLE METRICS DESCRIBING COMMUNITY STRUCTURE

Group attributes	
Group Size	The number of individuals (community members and visitors) in a group.
Community attributes	
Span	Total span of time steps that a community exists: the last time step – the first time step of the community's existence
Community Size	The number of individuals (members and absents but not visitors) affiliated with the community, averaged over the time steps a community is present
Individual attributes	
Switchiness	The number of community switches made by an individual in a population(normalized by the number of time steps an individual is observed)
Absenteeism	The number of absences of an individual from a community in a population(normalized by the number of time steps an individual is observed)
Visiting	The number of visits made by an individual to another community in a population (normalized by the number of time steps an individual is observed)
Average stay	The average number of consecutive time steps an individual stays affiliated with the same community over the time steps it is observed
Maximum stay	The maximum number of consecutive time steps an individual stays affiliated with the same community over the time steps it is observed
Community stay	The average number of consecutive time steps an individual stays affiliated with the same community over the time steps it is observed
Observed	Number of nodes observed in each time stamp.
Group size normalized	Group size normalized by the number of observed nodes.
Community size normalized	Community size normalized by the number of observed nodes.

7.1 Test aging-related changes as diminished intracortical connectivity

Given the results obtained from Louvain: the fact that communities are strongly evident only for young brains, and from the CommDy robustness analysis: which shows that old communities are less stable than younger, the leading hypothesis from a neuroscience prospective is the inability of old brains to synchronize effectively as young brains. For this reason an experiment has been build in which the same pipeline is adopted to identify communities in the brain but different levels of a drug (NMDA blocker APV, known to slow down synapses) has been injected in the brain slices. The purpose of this set up is to compare slow synapses to old brains and fast synapses to young brains.

Five different levels of drug dose have been used: 0, 15, 30, 60 ,120 µmol APV respectively with the following number of sample brains(datapoints): 10, 14, 7, 7, 2 for a total of 40 samples. The training data and have been divided into two groups : 0 and 15 representing the Young category (given the low amount of drug) (24 samples) and 30, 60 and 120 representing the Old category (16 samples). With this set up the majority class is used as baseline with accuracy 0.6.

Six types of classifiers are build in order to obtain a wide prospective on the potentiality of the dataset: Random Forest (n = 50 trees), Support Vector Machine (SVM) (linear kernel, decision function shape = one-vs-rest), Gaussian Naive Bayes, Maximum Entropy (solver = L-BFGS), AdaBoost (base estimator = classification tree depth 1, number of estimators = 50), XGBoost, and the classification is performed for each set of Commdy costs.

In order to verify the upperbound in the accuracy of the model a dataset made by both drugged data and the test data (composed of young and aged mice) has been made and the algorithm trained with 10-fold cross-validation, the results are consistent for each combination of commdy costs and are shown in Figure 41. The best accuracy is reached for the choice of costs 113, when XGBoost reaches 0.92 accuracy. It can also be seen that while XGBoost performs better with low Switching costs, AdaBoost performs better when the swithing cost is high.

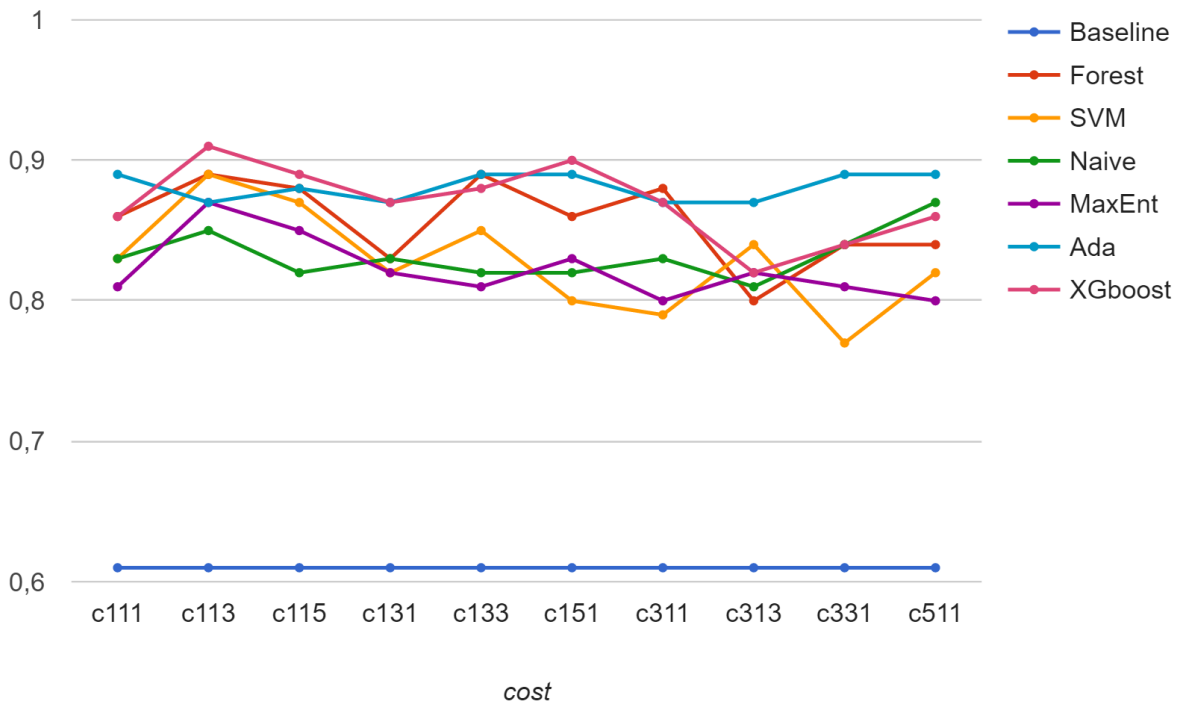


Figure 41: Accuracy when trained on APV-drug + Young-Old dataset, $\sigma < 0.1$ for each. Costs: Switching, Absence, Visiting.

Figure 42 shows the accuracy of the same classifiers trained on the drugged dataset and tested on the ground truth dataset (young and ages mice). The accuracy scores are obtained by running 20 times each algorithm. The graph shows that XGBoost has almost always the best performances and again this happens every time the visiting cost is higher (or equal) than the switching cost as in the previous case. Conversely AdaBoost does not perform very well in this scenario. It is also noticeable that by increasing the absence cost the accuracy of the random forest and the XGboost drops. The higher accuracy (0.84) is obtained with the cost set 133 by XGBoost.

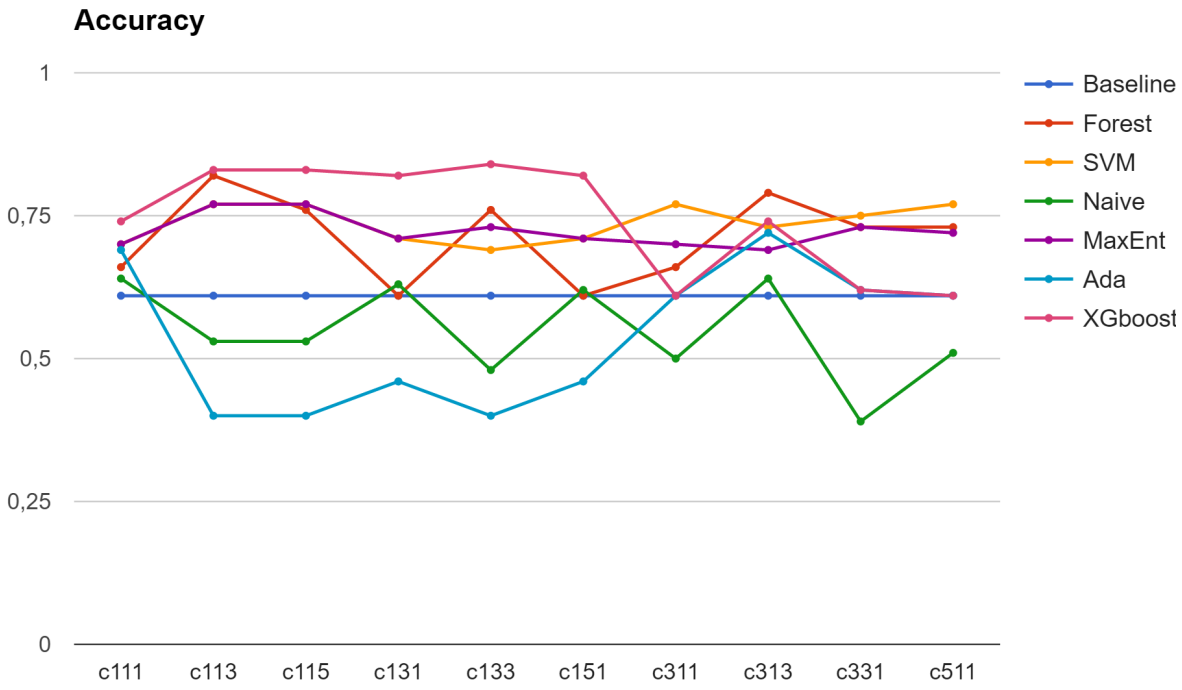


Figure 42: Accuracy when trained on APV-drug and tested on Young-Old dataset, σ random forest < 0.004. Costs: Switching, Absence, Visiting.

CHAPTER 8

CONCLUSION

In this work a pipeline to analyze auto-fluorescence brain mouse data is presented and each of the steps are validated. The result at each step give meaningful insight on the function of the method itself as well as present results in terms of identification of biological different communities.

The analysis shows how to choose the parameters to build a correlation network from brain data, proves the robustness and significance of the Louvain algorithm adopted to such dataset and highlights meaningful timestamp where communities are found.

The Dynamic Communities algorithm is evaluated in terms of robustness and significance, exploring the parameter space and evaluating the impact of different cost sets of the communities identified. The results are used to build a classifier that reaches an accuracy of 0.92 over a baseline of 0.6. Finally, a biological sound hypothesis is tested, as a result of the insights provided by the method, and the classification task is used to support it with a maximum accuracy of 0.82.

The analysis presented validates the use of the pipeline for fluorescence data without making any assumption on the underlying dataset, this choice has been made in order to let the data guide the investigation instead of ones hypothesis and beliefs. This is much more important since brain functionality is still mostly unknown and no assumption has to be made on the kind of outcomes to expect. For this reason most of the pipeline is not efficient and analyzes data

which, in the end, has been discovered to be unimportant (for instance all those pixels which hardly change value during the timeline). It is also noticeable that by relying on data, the pipeline has been tested for a further grade of overall robustness, which is the behavior when noise is inserted in the analysis, showing that overall the method can absorb noise and reveals key dynamic communities. This opens the road for further exploratory highly innovative studies which, cannot make any assumption because the system operation is unknown, and could adopt this pipeline to explore the presence of dynamic communities in their dataset. Finally, the result of this approach has been also an enormous effort in terms of computational power as well as being extremely time consuming (each analysis could take weeks).

CHAPTER 9

FUTURE WORK

The results of this work represent an exploratory work on the use of the presented pipeline to analyzed brain imaging data. Anyway this method has to be generalized to different datasets to be considered flexible and robust and, as stated in the conclusion chapter, the robustness to noise highly encourage the use of the pipeline with any dataset.

A second point of interest is the fact that the analysis has been carried on using the Pearson correlation as a base for the network construction, Dynamic Time Warping represents a much more effective way to compare time series and has not been used because infeasible with the amount of data to be processed. Anyway, having discovered which timestamps are of interest, it is possible and suggested the use of DTW to carry on the analysis and replace the Pearson correlation.

Finally, it has to be mentioned that the Louvain method has been the choice for this pipeline because much more effective on this dataset than the Infomap community detection algorithm. Anyway, this result, is highly dependent on the dataset chosen and the consequent correlation network which is build, for which topology matters more than flow. An other dataset may behave differently, thus it is valuable to make this test any time a new dataset is approached.

CITED LITERATURE

1. Sporns, O., Chialvo, D. R., Kaiser, M., and Hilgetag, C. C.: Organization, development and function of complex brain networks. Trends in cognitive sciences, 8(9):418–425, 2004.

2. Sporns, O.: Networks of the Brain. MIT press, 2011.

3. Menon, V.: Large-scale brain networks and psychopathology: a unifying triple network model. Trends in cognitive sciences, 15(10):483–506, 2011.

4. Llano, D. A., Theyel, B. B., Mallik, A. K., Sherman, S. M., and Issa, N. P.: Rapid and sensitive mapping of long-range connections in vitro using flavoprotein autofluorescence imaging combined with laser photostimulation. Journal of neurophysiology, 101(6):3325–3340, 2009.

5. Sporns, O.: Contributions and challenges for network models in cognitive neuroscience. Nature neuroscience, 17(5):652–660, 2014.

6. Robinson, L. F., Atlas, L. Y., and Wager, T. D.: Dynamic functional connectivity using state-based dynamic community structure: method and application to opioid analgesia. NeuroImage, 108:274–291, 2015.

7. Tantipathananandh, C., Berger-Wolf, T., and Kempe, D.: A framework for community identification in dynamic social networks. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 717–726. ACM, 2007.

8. Sekiyama, Y. and Kikuchi, J.: Towards dynamic metabolic network measurements by multi-dimensional nmr-based fluxomics. Phytochemistry, 68(16):2320–2329, 2007.

9. Chen, B., Fan, W., Liu, J., and Wu, F.-X.: Identifying protein complexes and functional modules—from static ppi networks to dynamic ppi networks. Briefings in bioinformatics, page bbt039, 2013.

CITED LITERATURE (continued)

10. Yosef, N., Shalek, A. K., Gaublomme, J. T., Jin, H., Lee, Y., Awasthi, A., Wu, C., Karwacz, K., Xiao, S., Jorgolli, M., et al.: Dynamic regulatory network controlling th17 cell differentiation. Nature, 496(7446):461–468, 2013.

11. Romanuk, T. N., Zhou, Y., Brose, U., Berlow, E. L., Williams, R. J., and Martinez, N. D.: Predicting invasion success in complex ecological networks. Philosophical Transactions of the Royal Society of London B: Biological Sciences, 364(1524):1743–1754, 2009.

12. Flack, J. C., Girvan, M., De Waal, F. B., and Krakauer, D. C.: Policing stabilizes construction of social niches in primates. Nature, 439(7075):426–429, 2006.

13. Kerth, G., Perony, N., and Schweitzer, F.: Bats are able to maintain long-term social relationships despite the high fission–fusion dynamics of their groups. Proceedings of the Royal Society of London B: Biological Sciences, 278(1719):2761–2767, 2011.

14. Petanidou, T., Kallimanis, A. S., Tzanopoulos, J., Sgardelis, S. P., and Pantis, J. D.: Long-term observation of a pollination network: fluctuation in species and interactions, relative invariance of network structure and implications for estimates of specialization. Ecology Letters, 11(6):564–575, 2008.

15. Norman, K. A., Polyn, S. M., Detre, G. J., and Haxby, J. V.: Beyond mind-reading: multi-voxel pattern analysis of fmri data. Trends in cognitive sciences, 10(9):424–430, 2006.

16. Babaei, M., Grabowicz, P., Valera, I., Gummadi, K., and Gomez-Rodriguez, M.: On the efficiency of the information networks in social media. In WSDM '16: Proceedings of the 9th ACM International Conference on Web Search and Data Mining, 2016.

17. Bullmore, E. and Sporns, O.: Complex brain networks: graph theoretical analysis of structural and functional systems. Nature Reviews Neuroscience, 10(3):186–198, 2009.

18. Rubinov, M. and Sporns, O.: Complex network measures of brain connectivity: uses and interpretations. Neuroimage, 52(3):1059–1069, 2010.

19. Betzel, R. F., Byrge, L., He, Y., Goñi, J., Zuo, X.-N., and Sporns, O.: Changes in structural and functional connectivity among resting-state networks across the human lifespan. Neuroimage, 102:345–357, 2014.

CITED LITERATURE (continued)

20. Achard, S. and Bullmore, E.: Efficiency and cost of economical brain functional networks. PLoS Comput Biol, 3(2):e17, 2007.

21. Achard, S., Salvador, R., Whitcher, B., Suckling, J., and Bullmore, E.: A resilient, low-frequency, small-world human brain functional network with highly connected association cortical hubs. The Journal of neuroscience, 26(1):63–72, 2006.

22. Hagmann, P., Cammoun, L., Gigandet, X., Meuli, R., Honey, C. J., Wedeen, V. J., and Sporns, O.: Mapping the structural core of human cerebral cortex. PLoS Biol, 6(7):e159, 2008.

23. Zuo, X.-N., Ehmke, R., Mennes, M., Imperati, D., Castellanos, F. X., Sporns, O., and Milham, M. P.: Network centrality in the human functional connectome. Cerebral cortex, 22(8):1862–1875, 2012.

24. van den Heuvel, M. P. and Sporns, O.: Rich-club organization of the human connectome. The Journal of neuroscience, 31(44):15775–15786, 2011.

25. Meunier, D., Lambiotte, R., Fornito, A., Ersche, K. D., and Bullmore, E. T.: Hierarchical modularity in human brain functional networks. Hierarchy and dynamics in neural networks, 1:2, 2010.

26. Meunier, D., Lambiotte, R., and Bullmore, E. T.: Modular and hierarchically modular organization of brain networks. Frontiers in neuroscience, 4:200, 2010.

27. Bassett, D. S., Greenfield, D. L., Meyer-Lindenberg, A., Weinberger, D. R., Moore, S. W., and Bullmore, E. T.: Efficient physical embedding of topologically complex information processing networks in brains and computer circuits. PLoS Comput Biol, 6(4):e1000748, 2010.

28. Bullmore, E. and Sporns, O.: The economy of brain network organization. Nature Reviews Neuroscience, 13(5):336–349, 2012.

29. Sporns, O. and Betzel, R. F.: Modular brain networks. Annual review of psychology, 67:613–640, 2016.

30. humanconnectomeproject.org: Human connectome project. Retrived October 3rd, 2016, from `http://http://www.humanconnectomeproject.org/`.

CITED LITERATURE (continued)

31. Hutchison, R. M., Womelsdorf, T., Allen, E. A., Bandettini, P. A., Calhoun, V. D., Corbetta, M., Della Penna, S., Duyn, J. H., Glover, G. H., Gonzalez-Castillo, J., et al.: Dynamic functional connectivity: promise, issues, and interpretations. Neuroimage, 80:360–378, 2013.

32. Handwerker, D. A., Roopchansingh, V., Gonzalez-Castillo, J., and Bandettini, P. A.: Periodic changes in fmri connectivity. Neuroimage, 63(3):1712–1719, 2012.

33. de Pasquale, F., Della Penna, S., Sporns, O., Romani, G., and Corbetta, M.: A dynamic core network and global efficiency in the resting human brain. Cerebral Cortex, page bhv185, 2015.

34. Liao, X., Yuan, L., Zhao, T., Dai, Z., Shu, N., Xia, M., Yang, Y., Evans, A., and He, Y.: Spontaneous functional network dynamics and associated structural substrates in the human brain. Frontiers in human neuroscience, 9, 2015.

35. Mizutani, R., Saiga, R., Takeuchi, A., Uesugi, K., and Suzuki, Y.: Three-dimensional network of drosophila brain hemisphere. Journal of structural biology, 184(2):271–279, 2013.

36. Huettel, S. A., Song, A. W., and McCarthy, G.: Functional magnetic resonance imaging, volume 1. Sinauer Associates Sunderland, 2004.

37. Hämäläinen, M., Hari, R., Ilmoniemi, R. J., Knuutila, J., and Lounasmaa, O. V.: Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain. Reviews of modern Physics, 65(2):413, 1993.

38. Niedermeyer, E. and da Silva, F. L.: Electroencephalography: basic principles, clinical applications, and related fields. Lippincott Williams & Wilkins, 2005.

39. Le Bihan, D., Mangin, J.-F., Poupon, C., Clark, C. A., Pappata, S., Molko, N., and Chabriat, H.: Diffusion tensor imaging: concepts and applications. Journal of magnetic resonance imaging, 13(4):534–546, 2001.

40. Llano, D. A., Turner, J., and Caspary, D. M.: Diminished cortical inhibition in an aging mouse model of chronic tinnitus. The Journal of Neuroscience, 32(46):16141–16148, 2012.

CITED LITERATURE (continued)

41. Eklund, A., Nichols, T. E., and Knutsson, H.: Cluster failure: Why fmri inferences for spatial extent have inflated false-positive rates. Proceedings of the National Academy of Sciences, page 201602413, 2016.

42. Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E.: Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment, 2008(10):P10008, 2008.

43. Meilă, M.: Comparing clusterings—an information based distance. Journal of multivariate analysis, 98(5):873–895, 2007.

44. Hubert, L. and Arabie, P.: Comparing partitions. Journal of classification, 2(1):193–218, 1985.

45. Rand, W. M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association, 66(336):846–850, 1971.

46. Fowlkes, E. B. and Mallows, C. L.: A method for comparing two hierarchical clusterings. Journal of the American statistical association, 78(383):553–569, 1983.

47. Rosvall, M. and Bergstrom, C. T.: Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences, 105(4):1118–1123, 2008.

48. Bohlin, L., Edler, D., Lancichinetti, A., and Rosvall, M.: Community detection and visualization of networks with the map equation framework. In Measuring Scholarly Impact, pages 3–34. Springer, 2014.

49. Fortunato, S.: Community detection in graphs. Physics reports, 486(3):75–174, 2010.

50. Fortunato, S. and Barthelemy, M.: Resolution limit in community detection. Proceedings of the National Academy of Sciences, 104(1):36–41, 2007.

51. Wasserman, S. and Faust, K.: Social network analysis: Methods and applications, volume 8. Cambridge university press, 1994.

52. Husson, T. R., Mallik, A. K., Zhang, J. X., and Issa, N. P.: Functional imaging of primary visual cortex using flavoprotein autofluorescence. The Journal of neuroscience, 27(32):8665–8675, 2007.

CITED LITERATURE (continued)

53. Sirotin, Y. B. and Das, A.: Spatial relationship between flavoprotein fluorescence and the hemodynamic response in the primary visual cortex of alert macaque monkeys. Front Neuroenergetics, 2(6), 2010.

54. Kitaura, H. and Kakita, A.: Optical imaging of human epileptogenic tissues in vitro. Neuropathology, 33(4):469–474, 2013.

55. Michael, N., Bischof, H.-J., and Löwel, S.: Flavoprotein autofluorescence imaging of visual system activity in zebra finches and mice. PloS one, 9(1):e85225, 2014.

56. Shibuki, K., Hishida, R., Murakami, H., Kudoh, M., Kawaguchi, T., Watanabe, M., Watanabe, S., Kouuchi, T., and Tanaka, R.: Dynamic imaging of somatosensory cortical activity in the rat visualized by flavoprotein autofluorescence. The Journal of physiology, 549(3):919–927, 2003.

57. Llano, D. A. and Sherman, S. M.: Differences in intrinsic properties and local network connectivity of identified layer 5 and layer 6 adult mouse auditory corticothalamic neurons support a dual corticothalamic projection hypothesis. Cerebral Cortex, page bhp050, 2009.

58. Crofoot, M. C., Rubenstein, D. I., Maiya, A. S., and Berger-Wolf, T. Y.: Aggression, grooming and group-level cooperation in white-faced capuchins (cebus capucinus): insights from social networks. American Journal of Primatology, 73(8):821–833, 2011.

59. Barale, C. L., Kulahci, I., Habiba, R. S., BergerYWolf, T., and Rubenstein, D. I.: A social networks approach to sheep movement and leadership. In 7 th International Conference on Applications of Social Network Analysis ASNA 2010, page 15. Citeseer, 2010.

60. Berger-Wolf, T., Fischhoff, I. R., Rubenstein, D. I., Sundaresan, S. R., and Tantipathananandh, C.: Dynamic analysis of social networks of equids. In 7 th International Conference on Applications of Social Network Analysis ASNA 2010, page 18. Citeseer, 2010.

61. Pearson, M. and West, P.: Drifting smoke rings. Connections, 25(2):59–76, 2003.

62. Leskovec, J., Lang, K. J., and Mahoney, M.: Empirical comparison of algorithms for network community detection. In Proceedings of the 19th international conference on World wide web, pages 631–640. ACM, 2010.

CITED LITERATURE (continued)

63. Backstrom, L., Huttenlocher, D., Kleinberg, J., and Lan, X.: Group formation in large social networks: membership, growth, and evolution. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 44–54. ACM, 2006.

64. Tantipathananandh, C. and Berger-Wolf, T.: Constant-factor approximation algorithms for identifying dynamic communities. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 827–836. ACM, 2009.

65. Müller, M.: Dynamic time warping. Information retrieval for music and motion, pages 69–84, 2007.

66. Berndt, D. J. and Clifford, J.: Using dynamic time warping to find patterns in time series. In KDD workshop, volume 10, pages 359–370. Seattle, WA, 1994.

67. Keogh, E. and Ratanamahatana, C. A.: Exact indexing of dynamic time warping. Knowledge and information systems, 7(3):358–386, 2005.

68. Sakoe, H. and Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE transactions on acoustics, speech, and signal processing, 26(1):43–49, 1978.

69. Sun, F. T., Miller, L. M., and D'Esposito, M.: Measuring interregional functional connectivity using coherence and partial coherence analyses of fmri data. Neuroimage, 21(2):647–658, 2004.

70. Silver, M. A., Landau, A. N., Lauritzen, T. Z., Prinzmetal, W., and Robertson, L. C.: Isolating human brain functional connectivity associated with a specific cognitive process. In IS&T/SPIE Electronic Imaging, pages 75270B–75270B. International Society for Optics and Photonics, 2010.

71. Lein, E. S., Hawrylycz, M. J., Ao, N., Ayres, M., Bensinger, A., Bernard, A., Boe, A. F., Boguski, M. S., Brockway, K. S., Byrnes, E. J., et al.: Genome-wide atlas of gene expression in the adult mouse brain. Nature, 445(7124):168–176, 2007.

72. Chen, J. E., Chang, C., Greicius, M. D., and Glover, G. H.: Introducing co-activation pattern metrics to quantify spontaneous brain network dynamics. NeuroImage, 111:476–488, 2015.

CITED LITERATURE (continued)

73. Berger-Wolf, T., Fischhoff, I. R., Rubenstein, D. I., Sundaresan, S. R., and Tantipathananandh, C.: Dynamic analysis of social networks of equids. In 7 th International Conference on Applications of Social Network Analysis ASNA 2010, page 18. Citeseer, 2010.

VITA

NAME	Umberto Di Fabrizio
EDUCATION	
	Master of Science in Computer Science, University of Illinois at Chicago, May 2015, USA
	Bachelor's Degree in Computer Engineering Polytechnic of Milan, 2014, Italy
LANGUAGE SKILLS	
Italian	Native speaker
English	Full working proficiency
SCHOLARSHIPS	
Fall 2015	Research Assistantship (RA) position (20 hours/week) with full tuition waiver
Spring 2015	Italian scholarship for TOP-UIC students
TECHNICAL SKILLS	
	Java, Python, MySQL, MongoDB, C, Javascript, R, Matlab, J2EE, Glassfish, Git, Linux.
WORK EXPERIENCE AND PROJECTS	
May 2016 - Aug 2016	Software Engineer Intern, Illumina, San Diego (CA)
Aug 2014 - Nov 2015	Data Visualization, Visualize and compare demographic data around the world
Nov 2015	Protein Superfamily Classification
Feb 2016	SpartaHack USAA Award Winner