Full title: A hands-on tutorial on network and topological neuroscience

Short title: Hands-on network and topological neuroscience

Eduarda Gervini Zampieri Centeno^{1,2}, Giulia Moreni¹, Chris Vriend^{1,3}, Linda Douw¹, Fernando
Antônio Nóbrega Santos^{1*}

¹Amsterdam UMC, Vrije Universiteit Amsterdam, Anatomy and Neurosciences, Amsterdam
Neuroscience, De Boelelaan 1117, Amsterdam, Netherlands;

²Bordeaux Neurocampus, Université de Bordeaux, Institut des Maladies Neurodégénératives,
CNRS UMR 5293, 146 Rue Léo Saignat, Bordeaux, France;

³Amsterdam UMC, Vrije Universiteit Amsterdam, Psychiatry, Amsterdam Neuroscience, De
Boelelaan 1117, Amsterdam, Netherlands;

* Corresponding Author:

Email: f.nobregasantos@amsterdamumc.nl

Authors Contributions:

EGZC contributions encompass **Conceptualisation, Software, Writing – Original Draft Preparation**

GM contributions encompass **Software, Writing – Original Draft Preparation**

CV contribution encompasses **Writing – Review & Editing**

LD contributions encompass **Conceptualisation, Writing – Original Draft Preparation, Supervision** FANS contributions
encompass **Conceptualisation, Software, Writing – Original Draft Preparation, Supervision**

Abstract

The brain is an extraordinarily complex system that facilitates the efficient integration of information from different regions to execute its functions. With the recent advances in technology, researchers can now collect enormous amounts of data from the brain using neuroimaging at different scales and from numerous modalities. With that comes the need for sophisticated tools for analysis. The field of network neuroscience has been trying to tackle these challenges, and graph theory has been one of its essential branches through the investigation of brain networks. Recently, topological data analysis has gained more attention as an alternative framework by providing a set of metrics that go beyond pair-wise connections and offer improved robustness against noise. In this hands-on tutorial, our goal is to provide the computational tools to explore neuroimaging data using these frameworks and to facilitate their accessibility, data visualisation, and comprehension for newcomers to the field. We will start by giving a concise (and by no means complete) overview of the field to introduce the two frameworks, and then explain how to compute both well-established and newer metrics on resting-state functional magnetic resonance imaging. We use an open-source language (Python) and provide an accompanying publicly available Jupyter Notebook that uses data from the 1000 Functional Connectomes Project. Moreover, we would like to highlight one part of our notebook that is solely dedicated to realistic visualisation of high order interactions in brain networks. This pipeline provides three-dimensional (3-D) plots of pair-wise and higher-order interactions projected in a brain atlas, a new feature tailor-made for network neuroscience.

47 Introduction

48 Neuroscience is still a young research field, with its emergence as a formal discipline
 49 happening only around 70 years ago [1]. The field has since mushroomed, and much of our
 50 current knowledge about the human brain's neurobiology was made possible by the rapid
 51 advances in technologies to investigate the brain in vivo at high-resolution and at different scales.
 52 An example is magnetic resonance imaging (MRI), which allows us not only to measure regional
 53 characteristics of the brain's structure non-invasively but may also be used to assess anatomical
 54 and functional interactions between brain regions [2, 3]. This expansion in the field led to an
 55 exponential increase in data size and complexity. To analyse and interpret this 'big data',
 56 researchers had to develop robust theoretical frameworks. Complex network science was
 57 brought to neuroscience and has been increasingly used to study the brain's intricate
 58 communication and wiring [4, 5]. The resulting field - network neuroscience – aims to see the
 59 brain through an integrative lens by mapping and modelling its elements and interactions [4, 6].

60 One of the main theoretical frameworks from complex network science used to model,
 61 estimate, and simulate brain networks is graph theory [7, 8]. A graph is comprised of a set of
 62 interconnected elements, also known as *vertices* and *edges*. Vertices (also known as nodes) in
 63 a network can, for example, be brain areas, while edges (also known as links) are a
 64 representation of the connectivity between pairs of vertices [5]. Various imaging modalities are
 65 available for the reconstruction of the brain network [8, 9]. The focus of this hands-on paper will
 66 be resting-state functional MRI (rsfMRI). As the name suggests, rsfMRI indirectly measures brain
 67 activity while a subject is at rest (*i.e.*, does not perform any task). This type of data provides
 68 information about spontaneous brain functional connectivity [10]. Connectivity is often

operationalised by a statistical dependency (usually a Pearson correlation coefficient) between signals measured from anatomically separated brain areas [2, 11].

Several descriptive graph metrics can then be calculated to describe the brain network's characteristic; examples include the degree or the total number of connections of a vertex and the path length (number of intermediate edges) between two vertices [6, 12]. These metrics have consistently allowed researchers to identify non-random features of brain networks. A key example is the ground-breaking discovery that the brain (like most other real-world networks) follows a 'small-world network' architecture [4, 13, 14]. This refers to the phenomenon that, to minimise wiring cost while simultaneously maintaining optimal efficiency and robustness against perturbation, the brain network obeys a balance between the ability to perform local processing (*i.e.*, segregation) and combining of information streams on a global level (*i.e.*, integration).

Network neuroscience has thereby offered a comprehensive set of analytical tools to study not only the local properties of brain areas but also their significance for the entire brain network functioning. Using graph theory, many insights have been gathered on the healthy and diseased brain neurobiology [5, 9, 12, 15].

Another perspective on the characteristics of the brain network can be provided by (algebraic) topological data analysis (TDA), by analysing the interactions between a set of vertices beyond the 'simple' pair-wise connections (*i.e.*, higher-order interactions). With TDA, one can identify a network's 'shape' and its invariant properties (*i.e.*, coordinate and deformation invariances [16, 17]). Thus, as we will illustrate along with the manuscript, TDA often provides more robustness against noise than graph theoretical analysis [18, 19], which can be a significant issue in imaging data [20-22]. Although TDA has only recently been adopted to network neuroscience, it has already shown exciting results on rsfMRI [23, 24]. For example,

group-level differences in network topology have been identified between healthy subjects that ingested psilocybin (psychedelic substance) and the placebo group [25], and between attention-deficit/hyperactivity disorder children and typically developing controls [26]. A downside of this framework is that the complexity and level of abstraction necessary to apply TDA and interpret the results might keep neuroscientists without prior theoretical training from using it. Moreover, the high-order interaction structure that emerges from TDA analysis is often difficult to visualise in a realistic and understandable manner.

Therefore, we would like to facilitate the use of network neuroscience and its constituents graph theory and TDA in the general neuroscientific community by providing a step-by-step tutorial on how to compute different metrics commonly used to study brain networks as well as realistic high-order network visualisation tools. We provide a theoretical and experimental background of these metrics and include code blocks in each section to explain how to compute the different metrics. We also list several additional resources (Table 1 and Table 2) of personal preference (and by no means complete), including a Jupyter Notebook that we created to accompany this hands-on tutorial publicly available on GitHub and Zenodo [27] (see Table 1, under the Jupyter Notebooks section - Notebook for network and topological analysis in neuroscience).

110 **Table 1. List of computational resources.**

	Name	Brief explanation	Source
Jupyter Notebooks			
	DyNeuSR	Notebook on how to use Mapper - an algorithm for high dimensional dataset exploration.	https://github.com/braindynamicslab/dyneusr-notebooks/
	Notebook for network and topological analysis in neuroscience	Notebook on how to compute both classical and newer metrics of network and topological neuroscience.	https://github.com/multinetlab-amsterdam/network_TDA_tutorial
	Tutorials for Topological Data Analysis with the Gudhi Library	A collection of notebooks for the practice TDA with the python Gudhi library.	https://github.com/GUDHI/TDA-tutorial/
	AML-days-TDA-tutorial	A set of notebooks on the theory and applications of TDA pipelines.	https://github.com/lordgrilo/AML-days-TDA-tutorial
MATLAB toolboxes and scripts			
	CliqueTop	A collection of MATLAB scripts for TDA.	https://github.com/nebneuron/clique-top
	The brain connectivity toolbox	MATLAB toolbox for brain network analysis.	https://sites.google.com/site/bctnet/
Python packages and scripts			
Data visualisation	DyNeuSR	"DyNeuSR is a Python visualisation library for topological representations of neuroimaging data."	https://braindynamicslab.github.io/dyneusr/
	Nxviz	"nxviz is a graph visualisation package for NetworkX."	https://nxviz.readthedocs.io/
	Plotly	"Plotly's Python graphing library makes interactive, publication-quality graphs."	https://plot.ly/python/
Graph theory	Bctpy	"A direct translation to Python of the MATLAB brain connectivity toolbox."	https://github.com/aestrivex/bctpy
	NetworkX	"A package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks."	https://networkx.github.io/
TDA			
	Dionysus	"A library for computing persistent homology. It is written in C++, with Python bindings."	https://mrzv.org/software/dionysus2/
	Giotto	"A collection of algorithms that harbours theoretical and technological advances spanning several key disciplines, including TDA."	https://giotto.ai/
	Gudhi	"The library offers state-of-the-art data structures and algorithms to construct simplicial complexes and compute persistent homology."	http://gudhi.gforge.inria.fr/
	Scikit-TDA	"Topological Data Analysis Python libraries intended for non-topologists."	https://scikit-tda.org/
	The brain connectivity toolbox	MATLAB toolbox for brain network analysis.	https://sites.google.com/site/bctnet/

111 Table content was organised in alphabetic order.

112

113 **Table 2. List of reading resources.**

Name	Source
Key articles and books	
Cliques and cavities in the human connectome	Reference [3]
Network neuroscience	Reference [4]
Fundamentals of brain network analysis (<i>the primary reference of our hands-on tutorial</i>)	Reference [6]
Graph theory approaches to functional network organisation in brain disorders: a critique for a brave new small-world	Reference [12]
Topology for computing	Reference [16]
The importance of the whole: Topological data analysis for the network neuroscientist	Reference [20]
Editorial: Topological Neuroscience	Reference [23]
What can topology tell us about the neural code?	Reference [24]
Homological scaffolds of brain functional networks	Reference [25]
Two's company, three (or more) is a simplex	Reference [28]
A roadmap for the computation of persistent homology	Reference [29]
Networks beyond pair-wise interactions: Structure and dynamics	Reference [30]
Clique topology reveals intrinsic geometric structure in neural correlations	Reference [31]

114 Table content was organised in numerical order.

115

116 Our work differs from previous literature [12, 29] since we not only describe the concepts
117 central to graph theory and TDA but also provide an easy-to-grasp step-by-step tutorial on how
118 to compute these metrics using an easily accessible, open-source computer language.
119 Furthermore, we offer new 3-D visualisations of simplicial complexes and TDA metrics in the
120 brain that may facilitate the application and interpretation of these tools. Finally, we would like to
121 stress that even though this tutorial focuses on rsfMRI, the main concepts and tools discussed
122 in this paper can be extrapolated to other imaging modalities, biological or complex networks.

123 Since graph theory has been extensively translated for neuroscientists elsewhere, we
124 refer the reader to the book in [6]. In this tutorial, we mainly focused on the topics covered in

chapters 3, 4, 5, and the particular sections of chapters 6, 7, 8 and 9 about assortativity, shortest paths and the characteristic path length, the clustering coefficient, and modularity. In the second part of the tutorial, we explore hands-on TDA metrics, providing a summary of both theoretical and neuroscientific aspects with the calculations used in our work. We believe that our tutorial, which is far from being exhaustive, can contribute to make this emerging branch of network and topological neuroscience accessible to the reader. The codes we provide usually only requires the knowledge of the connectivity matrix and, for our realistic 3d visualisation of simplicial complexes, we only need the coordinates of the nodes of a given brain atlas. Therefore, we believe that our scripts can be adapted to different databases, image modalities and brain atlas. A short glossary with the key terms to understand this manuscript can be found in Table 3.

Table 3. Glossary with key terms.

Term	Brief explanation
Clique	A subset of vertices in an undirected graph in which all vertices are connected to each other.
Clique complex	A simplicial complex constituted of all cliques of a network.
Clique participation rank	The number of k -cliques in which a vertex i participates for density d .
Connectivity matrix	A square $N \times N$ matrix is used to represent connectivity between vertices.
Face	A subset of a k -simplex. For example, if the k -simplex is a 2-simplex (triangle), all edges and vertices composing this simplex are also its faces.
Filtration	A nested sequence of simplicial complexes.
Functional magnetic resonance imaging (fMRI)	The imaging technique used to measure brain activity by detecting brain blood flow changes, <i>i.e.</i> , blood-oxygen-level-dependent (BOLD) signal.
k-simplex	Geometrically, it is the generalisation of the region delimited by a tetrahedron to an arbitrary dimension k , which can be done in many ways [16]. In this work, a k -simplex is a complete graph of $k+1$ vertices. For example, 0-simplex is a point (or vertex), 1-simplex is a line segment (or edge), 2-simplex is a triangle, and so on.

Simplicial complex

A simplicial complex K is a finite set of k -simplexes (e.g., vertices, edges, triangles, tetrahedrons, and their n -dimensional counterparts). The formal definition states that if K contains a k -simplex, then K also contains all faces of this k -simplex. Moreover, if two simplexes in K intersect, then this intersection is a face of each of them.

Table content was organised in alphabetic order.

Hands-on tutorial

General requirements

The following Python 3 packages are necessary to perform the computations presented below. The accompanying Jupyter Notebook can be found on GitHub (Table 1) or Zenodo [27].

Code example

```
# Import required Python packages
import numpy as np # version 1.18.5
import networkx as nx # version 2.4
import community # version 0.13 (python-louvain)
import gudhi # version 3.3.0
import scipy.io # version 1.4.1
from sklearn import preprocessing # version 0.23.1
import itertools
import seaborn as sns # version 0.11.0
import matplotlib.pyplot as plt # version 3.3.2
```

The basis: the adjacency matrix

The basic unit on which graph theory and TDA are applied in the context of rsfMRI in our work is the adjacency or connectivity matrix (Fig 1G), which present the connections between all vertices in the network [4-6, 32]. Typically, rsfMRI matrices do not specify the

direction of connectivity (*i.e.*, activity in area A drives activity in area B), therefore yielding undirected networks (Fig 1B and Fig 1F).

Before calculating any metrics on such matrices, several crucial factors must be considered when dealing with connectivity data [12, 33]. One critical decision is whether one wants to keep the information about edge weights. When the edges' weights (*e.g.*, correlation values in rsfMRI connectivity) are maintained, the network will be weighted (Fig 1D and Fig 1F). Another approach is to use a threshold, *e.g.*, only keep and binarise the 20% strongest connections (Fig 1A and Fig 1B). There is currently no gold standard for the weighting issue in rsfMRI matrices [6, 33] and may also be dependent on the dataset or proposed analysis [34].

Another relevant discussion about rsfMRI matrices is the interpretation of negative weights or anticorrelations. The debate of what such negative correlations mean in neurophysiology is still going on [35]. Studies have suggested that they could be considered artefacts introduced by global signal regression or pre-processing methods, or simply by large phase differences in synchronised signals between brain areas [36, 37]. Nevertheless, a few authors have suggested that anticorrelations might carry biological meaning underlying long-range synchronisation and that in diseased states alterations in these negative correlations could indicate network reorganisation [35, 36]. Negative weights can be absolutised to keep the potential biological information they may carry. If one decides to discard them, it is crucial to keep in mind that some physiological information might be lost [6, 12, 35, 36].

In this tutorial, we will use an undirected, absolutised (positively) weighted matrix.

Fig 1. Types of networks. (A) A binary directed graph. (B) Binary, undirected graph. In binary graphs, the presence of a connection is signified by a 1 or 0 otherwise. (C) A representation of graph F as a network of brain areas. (D) A weighted, directed graph. (F) A weighted, undirected graph. In a weighted graph, the absolute strength of the connections is often represented by a number w , where $0 \leq w \leq 1$. (G) A connectivity matrix of C and F. Source: Part of the image was obtained from smart.servier.com.

To follow the steps below, we assume that rsfMRI was already pre-processed and converted to a matrix according to some atlas. Steps and explanations on data pre-processing and atlas choices are beyond the scope of this paper, please see [38] for further information. Details on our Jupyter Notebook's dataset pre-processing can be found in [39, 40].

Code example

```
# Import matrix
matrix = np.genfromtxt('matrixexample.txt')

# Absolutize matrix
matrix = abs(matrix)
```

When working with fMRI brain network data, it is useful to generate some plots (e.g., the heatmaps for matrix visualisation, and distribution plots of edge weights) to facilitate data exploration, comprehension and flag potential artefacts. In brain networks, we expect mostly weak edges and a smaller proportion of strong ones. When plotted as a probability density of log10, we expect the weight distribution to have a Gaussian-like form [6].

Code example

```
# Visualize matrix with seaborn heatmap (Similar to Fig 1G)
matrixdiagNaN = matrix.copy()
np.fill_diagonal(matrixdiagNaN, np.nan) # Send diagonal to NaN for a clearer
hetmap
sns.heatmap(matrixdiagNaN, cmap='coolwarm', cbar=True, square=False)

# Weight distribution plot
bins = np.arange(np.sqrt(len(np.concatenate(matrixdiagNaN))))
bins = (bins - np.min(bins))/np.ptp(bins)
fig, axes = plt.subplots(1,2, figsize=(15,5))

# Distribution of raw weights
rawdist = sns.distplot(matrixdiagNaN.flatten(), bins=bins, kde=False,
ax=axes[0], norm_hist=True)
rawdist.set(xlabel='Correlation Values', ylabel = 'Density Frequency')

# Probability density of log10
log10dist = sns.distplot(np.log10(matrixdiagNaN).flatten(), kde=False,
ax=axes[1], norm_hist=True)
log10dist.set(xlabel='log(weights)')
```

Graph theory

Here, we will cover the most commonly used graph metrics in network neuroscience (see Fig 2), also in line with reference [6]. First, we need to create a graph object using the package *NetworkX* [41] and remove the self-loops (*i.e.*, the connectivity matrix's diagonal).

Code example

```
# Create graph from the matrix
G_gt = nx.from_numpy_matrix(matrix)

# Remove self-loops in the graph
G_gt.remove_edges_from(G_gt.selfloop_edges())
```

Fig 2. Graph theoretical metrics. (A) A representation of a graph indicating centralities. Highest degree centrality indicates the vertex with most connections. Highest betweenness centrality refers to the vertex with most short paths passing through it. Highest closeness centrality denotes the vertex that needs the least edges to reach all the other nodes. The highest eigenvector centrality is achieved by the vertex which is best connected to the rest of the network, considering the number of neighbours and how well connected they are. (B) Representation of modularity and clustering coefficient. The latter indicates the tendency for any two neighbours of a vertex to be directly connected to each other. (C) The shortest path between vertices A and B. (D) The minimum spanning tree is a subset of a graph's edges, which does not contain cycles, and that has the lowest sum of distances.

Degree:

Vertex *degree* quantifies the total number of vertex connections in an undirected binary network [6]. In an undirected weighted network like our rsfMRI matrix, the vertex

degree is analogous to the vertex *strength* (i.e., the sum of all edges of a vertex) and equivalent to its degree centrality. This metric is one of the most fundamental metrics in network analysis and is a useful summary of how densely individual vertices are connected. It can be computed as the sum of edge weights of the neighbours of vertex i :

$$C_D = s_i = \sum_{j \neq i} w_{ij},$$

where w_{ij} is the weight of the edge linking vertices i and j .

Code example

```
# Compute vertex strength for vertex number 1
G_gt.degree(nbunch=1, weight='weight')
```

By removing the argument *weight* from the function, one can compute the degree of binarised networks where all edges are either 0 or 1 (useful if working with a sparse/not fully connected matrix). This change will give the vertex degree by calculating the number of edges adjacent to the vertex. One can also remove the specified vertex to estimate the degree/strength of all vertices. The *degree/strength distribution* allows us to scope the general network organisation in a single shot by displaying whether the network contains a few highly connected vertices, i.e., hubs [12].

Path length:

- a) The *shortest path* is the path with least number of edges (or least total weight) between two vertices in a network. In a weighted graph, the shortest path is calculated by the minimum sum of the weights of edges between two vertices [6]. It is seen as a measure for understanding the efficiency of information diffusion in a network. Several algorithms can calculate path lengths, but Dijkstra's algorithm [42] is one of the oldest and most well-known. An important detail is that this algorithm is only applicable to graphs with non-negative weights [42].
- A pivotal point to keep in mind is that in the case of correlation matrices, such as rsfMRI data, the weights must be converted to 'distance' by computing the inverse of the original weight ($1 - \text{weight}$ or $\frac{1}{\text{weight}}$); a higher correlation value represents a shorter distance [6]. This conversion is essential for all the following path-based metrics.
- b) *Average path length (or characteristic path length)* is the average shortest path length for all possible pairs of vertices in a network. It is a global measure of information transport efficiency and integration in a network and is widely known due to the famous Watts-Strogatz model [14]. It can be computed as follows:

$$L = \sum_{i,j \in V} \frac{d(i,j)}{N(N-1)},$$

where V is a set of vertices, $d(i, j)$ is the shortest path between vertices i and j , and N is the number of vertices in the network.

Code example

```
# We calculate the distance as 1-weight
G_gt_distance_dict = {(e1, e2): 1 / abs(weight) for e1, e2, weight in
G_gt.edges(data='weight')}

# Then we need to add it as an attribute of the graph edges
nx.set_edge_attributes(G_gt, values= G_gt_distance_dict,
name='distance')

# Compute path length between vertices 1 and 5
nx.dijkstra_path_length(G_gt, source=1, target=5, weight='distance')

# Compute path length between all pairs
nx.all_pairs_dijkstra_path_length(G_gt, weight='distance')

# Compute average path length
nx.average_shortest_path_length(G_gt, weight='distance')
```

The clustering coefficient:

The clustering coefficient assesses the tendency for any two neighbours of a vertex to be directly connected (or more strongly connected in the weighted case) to each other, and can also be termed cliquishness [12, 14]. This metric is also used for the computation of the small-worldness coefficient (ratio between the characteristic path length and the clustering coefficient relative to random networks) [14]. The formula can be defined as:

$$Cl = \frac{2}{s_i(s_i - 1)} \sum_{j,h} (\hat{w}_{ij} \hat{w}_{jh} \hat{w}_{hi})^{\frac{1}{3}},$$

where s_i is the degree/strength of vertex i , and the edge weights' are normalised by the maximum weight in the network, such that $\hat{w}_{ij} = \frac{w_{ij}}{\max(w)}$.

Code example

```
# Compute clustering coefficient of the network
nx.clustering(G_gt, weight='weight')

# Compute the average clustering coefficient of the network
nx.average_clustering(G_gt, weight='weight')
```

Centralities:

a) *Eigenvector (degree-based) centrality* measures a vertex's importance in a network while also considering how important its neighbours are [43]. Thus, it considers both the quantity and quality of a vertex's connections. The eigenvector centrality can be computed as:

$$Ax = \lambda x,$$

where A is the adjacency matrix, and x is an eigenvector of A with eigenvalue λ . We can also see the eigenvector centrality of a vertex i as the summed centrality of its neighbours:

$$C_E(i) = \frac{1}{\lambda_1} \sum_{j=1}^N A_{ij} x_j.$$

362

363

364

365

366

367

368

369

370

371

372

373

374

$$Cc(i) = \frac{N - 1}{\sum_{j=1}^{N-1} d(i, j)},$$

375

376

377

378

379

380

381

382

383

For weighted networks, certain conditions apply. According to the Perron-Frobenius theorem, the adjacency matrix's largest eigenvalue must be unique and positive, which is guaranteed only for matrices with positive values [6, 44].

b) *Closeness (shortest path-based) centrality* is a measure of how closely or 'directly' connected a vertex is to the rest of the network. If the vertex is the closest to every other element in the network, it has the potential to spread information fast and efficiently [6]. Formally, the closeness centrality of a vertex i is the inverse of its average shortest path length (*N.B.* weights need to be converted to distances) to all $N-1$ other vertices:

where $d(i, j)$ is the shortest-path distance between i and j , and N is the number of vertices in the graph. In weighted networks, closeness centrality can be estimated by considering the summed weight of the shortest paths according to Dijkstra's algorithm [42].

c) *Betweenness (shortest path-based) centrality* is the proportion of all vertex-pairs shortest paths in a network that pass through a particular vertex [44, 45]. It is used to understand the importance of vertices in the overall flow of information in a

network. To compute the betweenness centrality of a vertex i , one has to calculate the proportion of shortest paths between two vertices, e.g., i, j , that pass through vertex h :

$$C_B(h) = \sum_{i \neq h \neq j \in V} \frac{\sigma_{ij}(h)}{\sigma_{ij}},$$

where V is a set of vertices, σ_{ij} is the total number of shortest paths between i and j , and $\sigma_{ij}(h)$ is the number of those paths that pass through h . For weighted graphs, edges must be greater than zero, and the metric considers the sum of the weights [6]. Again, it is necessary to use the distance when using this shortest path-based metric. This formula can also be normalised by putting $\frac{2}{(N-1)(N-1)}$ in front of the sum (N being the number of vertices).

Code example

```
# Compute eigenvector centrality for graph
nx.eigenvector_centrality(G_gt, weight='weight')
```

```
# Compute closeness centrality for graph using distance
nx.closeness_centrality(G_gt, distance='distance')

# Compute betweenness centrality for graph
nx.betweenness_centrality(G_gt, weight='distance', normalized=True)
```

The minimum spanning tree:

The minimum spanning tree is the backbone of a network, *i.e.*, the minimum set of edges necessary to ensure that paths exist between all vertices without forming cycles [46, 47]. A few main algorithms are used to build the spanning tree, with Kruskal's algorithm being implemented in *NetworkX* [48]. Briefly, this algorithm ranks the distances between vertices, adds the ones with the smallest distance first, and adding edge by edge it checks if cycles are formed or not. The algorithm will not add an edge that results in the formation of a cycle.

Code example

```
# Compute the minimum spanning tree
nx.minimum_spanning_tree(G_gt, weight='distance')
```

Modularity:

Modularity states how divisible a network is into different modules (or communities) and the identification of the modules is performed by the community detection algorithm [6, 8, 49]. Here, we will use the Louvain algorithm [50] as recommended by [6]. It works in a two-step iterative manner, first looking for communities by optimising modularity locally and then concatenating vertices that belong to the same module [50].

Code example

```
# Computing Modularity
part = community.best_partition(G_gt, weight='weight') ## Louvain
algorithm

# Check the number of communities
set(part.values()).union()
```

Topological data analysis

In this section, we will use TDA on our rsfMRI adjacency matrices. TDA can identify different characteristics of a network by addressing the high-order structure of a network beyond pair-wise connections as used in graph theory [30, 51, 52]. TDA generally uses topology and geometry methods to study the shape of the data [53]. A core success of TDA is the ability to provide robust results when compared with alternative methods, even if the data are noisy [18, 23]. One of the benefits of using TDA in network neuroscience is the possibility of finding global properties of a network that are preserved regardless of the way we represent the network [25], as we will illustrate below. Those properties are the so-called topological invariants.

We will cover some fundamental TDA concepts: filtration, simplicial complexes, Euler characteristic, phase-transitions, Betti numbers, curvature, and persistent homology. A summary can be found in Fig 3.

Fig 3. Topological data analysis. (A) Illustration of simplexes. (B) Representation of simplexes/cliques of different order being formed in the brain across the filtration process. (C) Barcode respective to panel B, representing the filtration across distances (*i.e.*, the inverse of weights in a correlation matrix). Line A represents cycle A in B. β_0, β_1 , and β_3 indicate the homology groups. (β_0 = connected components, β_1 = one-dimensional holes, β_2 = 2-dimensional holes). (D) Circular projection of how the brain would be connected. (E) Persistence diagram (or Birth/Death plot) obtained from real rsfMRI brain data. In this plot, it is also possible to identify a phase transition between β_1 and β_2 .

The basis: the adjacency matrix and filtration

As indicated in the earlier section on graph theory, there is no consensus on the necessity or level of thresholding performed on rsfMRI-based adjacency matrices. However, TDA overcomes this problem by investigating connectivity over all possible thresholds in a network. This process of investigating network properties looking for

all possible thresholds instead of choosing a fixed one is called *filtration* (Fig 3B and S1 Video). It consists of changing the threshold, e.g., the density d of the network, from $0 \leq d \leq 1$. This yields a nested sequence of networks, in which increasing d leads to a more densely connected network. Note, however, that the notion of filtration is not unique to TDA and has also been applied in graph-theoretical work [26, 54].

Code example

```
# Define function to create a graph with specific density
def graph_density(d, matrix):
    np.fill_diagonal(matrix,0)
    # Flatten and rank the correlation values
    temp = sorted(matrix.ravel(), reverse=True)
    size = len(matrix)
    cutoff = np.ceil(d*(size*(size-1)))
    value = temp[int(cutoff)]
    G0 = nx.from_numpy_matrix(matrix)
    G0.remove_edges_from(list(nx.selfloop_edges(G0)))
    G1 = nx.from_numpy_matrix(matrix)

    for u,v,a in G0.edges(data=True):
        if (a.get('weight')) <= value:
            G1.remove_edge(u, v)

    finaldensity = nx.density(G1)
    print(finaldensity) # To check the density of the new graph

    return G1

# Filtration from density 0 to 1 in steps of 0.02
for density in np.arange(0, 1, 0.02):
    graph_density(density, matrix)
```

Simplicial complexes

In TDA, we consider that the network as a multidimensional structure called the simplicial complex. Such a network is not only made up of the set of vertices (0-

simplex) and edges (1-simplex) but also of triangles (2-simplex), tetrahedrons (3-simplex), and higher k -dimensional structures (Fig 3A). In short, a k -simplex is an object in k -dimensions and, in our work, is formed by a subset of $k+1$ vertices of the network (Fig 4).

Fig 4. Simplicial complex. An example of a simplicial complex composed of eight vertices (0-simplexes), 11 edges (1-simplexes), five triangles (2-simplexes), one tetrahedron (3-simplexes).

We can encode a network into a simplicial complex in several ways [55-57]. However, here, we will focus on building a simplicial complex only from the brain network's cliques, *i.e.*, we will create the so-called clique complex of a brain network. In a network, a k -clique is a subset of the network with k all-to-all connected nodes. 0-clique corresponds to the empty set, 1-cliques correspond to nodes, 2-cliques to links, 3-cliques to triangles, and so on. In the clique complex, each $k+1$ clique is associated with a k -simplex. This choice for creating simplexes from cliques has the advantage that we can still use pair-wise signal processing to create a simplicial complex from brain networks, such as in [31]. It is essential to mention that other strategies to build simplicial complexes beyond pair-wise signal processing are still

under development, such as applications using multivariate information theory together with tools from algebraic topology [58-63].

In our Jupyter Notebook, we provide the code to visualise the clique complex developed in [64]. To create the 3-D plots, we used mesh algorithms available in *Plotly* [65], together with a mesh surface of the entire brain available in [66, 67]. In Fig 5, we display an example of 3-D visualisation of 3-cliques in the HCP data. When we increase the filtration density d , we obtain more connections, and more 3-cliques arise. In Fig 5, only 3-cliques are shown; however, the same can be done for higher-dimensional cliques like a tetrahedron, *et cetera*. In the S1 Video, we show a filtration in a functional brain network up to 4-cliques. In the Jupyter Notebook, we are also able to visualise the clique complex at arbitrary sizes, up to computational limits. The computation is not shown here as code blocks due to its size and complexity (see the Jupyter Notebook).

Fig 5. Simplex 3-D visualisation. Here we visualise the rising number of 3-cliques (triangles) in a functional brain network as we increase the edge density d (0.01, 0.015, 0.02, and 0.025, from A to D). For higher densities, we have a more significant number of 3-cliques compared to smaller densities. The vertex colour indicates the clique participation rank.

The Euler characteristic:

The Euler characteristic is one example of topological invariants: the network properties that do not depend on a specific graph representation. We first introduce the Euler characteristic for polyhedra, as illustrated in Fig. 6. Later, we translate this

concept to brain networks. In 3-D convex polyhedra (for example a cube, a tetrahedron, *et cetera*, see Fig. 6), the Euler characteristic is defined as the numbers of vertices minus edges plus faces of the considered polyhedra. For convex polyhedra without cavities (holes in its shape), which are isomorphous to the sphere, the Euler characteristic is always two, as you can see in Fig 6. If we take the cube and make a cavity, the Euler drops to zero as it is in the torus. If we make two cavities in a polyhedral (as in the bitorus), the Euler drops to minus two (Fig 7). We can understand that the Euler characteristic tells us something about a polyhedron's topology and its analogous surface. In other words, if we have a surface and we make a discrete representation of it (e.g., a surface triangulation), its Euler characteristic will always be the same, regardless of the way we do it.

<i>Name</i>	<i>Image</i>	<i>Vertices</i> <i>V</i>	<i>Edges</i> <i>E</i>	<i>Faces</i> <i>F</i>	<i>Euler Characteristic</i> $V - E + F$
<i>Tetrahedron</i>		4	6	4	2
<i>Cube</i>		8	12	6	2
<i>Octahedron</i>		6	12	8	2
<i>Dodecahedron</i>		20	30	12	2

Fig 6. Euler characteristic in convex polyhedra. Note that there are no cavities in their shapes for convex polyhedra, and the Euler characteristic is always equal to two.

<i>Cube</i>	<i>Sphere</i>	<i>Euler Characteristic</i>
2
<i>Cube with 1 cavity</i>	<i>Torus</i>
0
<i>Cube with 2 cavities</i>	<i>Bitorus</i>
-2

Fig 7. The Euler characteristic in polyhedra with cavities. The Euler characteristic of a cube with a cavity is equal to zero, just as the torus. This value drops to minus two if we have two cavities in the cube, just like a bitorus.

We can now generalise the definition of Euler characteristic to simplicial complex in any dimension. Thus, the high dimensional version of the Euler characteristic is expressed by the alternate sum of the numbers $Cl_k(d)$ of the k -cliques (which are $(k-1)$ -simplexes) present in the network's simplicial complex for a given value of the density threshold d .

$$\chi(d) = Cl_1 - Cl_2 + \dots + (-1)^n Cl_n = \sum_{k=1}^n (-1)^k Cl_k(d).$$

Code example

```
# Let us create a graph with a specific density using our function above
G_tda = graph_density(0.2, matrix))

# First Method
E = euler(G_tda, verbose=True)
print("Euler value using Euler function:", E)

# Second Method
S = euler_k(G_tda, 4, verbose=True)
print("Euler value (using maximum clique size of 4, i.e., k_max = 4): ",
      S[0])
```

Note that the clique algorithm, which is the primary function used in our code (*euler* - S1 Code Block), is an NP-complete problem, which is computationally expensive for large and/or dense networks, regardless of the way you implement it [68]. An alternative is to fix an upper bound for the cliques' size [68, 69]. Therefore, the second function (*euler_k* - S1 Code Block) allows the user to constrain the maximum size of the cliques we are looking for. This means that we are fixing the dimension k of our simplicial complex and ignoring simplexes of dimension greater than k .

Topological phase transitions:

Phase transitions can provide insight into the properties of a 'material'. For example, water is known for becoming steam at 100°C. Similarly, by using TDA when comparing a patient and healthy population, one could identify these populations' properties by studying the topological phase transition profile for each group. This strategy has already been applied for investigating group differences between controls and glioma brain networks [64], and typically developing children and children with attention-deficit/hyperactivity disorder [26]. In other fields, topological phase transitions

were also investigated in the *S. cerevisiae* and *C. elegans* protein interaction networks, reionization processes, and evolving coauthorship networks [70-72].

To investigate topological phase transitions in brain networks, we first need to visualise the *Euler entropy* (Fig 3 in [73]):

$$S_{\chi} = \ln |\chi|.$$

When $\chi = 0$ for a given value of the density of the network, the Euler entropy is singular, $S_{\chi} \rightarrow \infty$. Under specific hypotheses, a topological phase transition in a complex network occurs when the Euler characteristic is null [64]. This statement finds support in the behaviour of S_{χ} at the thermodynamic phase transitions across various physical systems [73]. In network theory, the Giant component transition is associated with network changes, from smaller connected clusters to the emergence of Giant ones [74]. Theoretically, topological phase transitions are related to the extension of the Giant component transition for simplicial complexes [75]. Based on numerical simulations, it was also conjectured that the longest cycle is born in the phase transition vicinity [76, 77]. Phase-transitions can also be visualised in Birth/Death plots (Fig 3E) which will be discussed later in the Persistent Homology section.

Betti numbers:

Another set of topological invariants are the Betti numbers (β). Given that a simplicial complex is a high-dimensional structure, β_k counts the number of k -dimensional holes in the simplicial complex. These are topological invariants that

correspond, for each $k \geq 0$, to the number of linearly independent k -dimensional holes in the simplicial complex [16].

In Fig 8, we show the representation of the k -dimensional holes. We give one example for each dimension. In a simplicial complex, there can be many of these k -holes and counting them provide the Betti number β , e.g., if β_2 is equal to five, there are 5 two dimensional holes.

Fig 8. Betti numbers and examples of each k -dimensional hole. β_0 is the number of connected components or zero-dimensional holes. β_1 is the number of one-dimensional holes (loops). β_2 is the number of two-dimensional holes (voids). β_3 is the number of 3-D holes. For $\beta_1, \beta_2, \beta_3$ only the left figure of each pair represents the k -dimensional hole. In the right figure, a connection is added, and so the k -hole is lost: the right figure of each pair no longer represent a β hole. For β_0 the number of connected components is the number of separate clusters we have in the figure; therefore, we should consider the figure as a whole (in the case represented here we have four connected components).

Code example

```
# Put the Betti number you want to compute
beti = 1

# Compute Betti number
B = Betti_k(G_tda, beti, verbose=True) # with verbose= True you can see
all the steps of the algorithm
```


Notice that for higher k or dense simplicial complexes, the calculation of the β becomes computationally expensive.

There are ways to estimate the β of a simplicial complex without calculating it directly. It is known that the β relate to Euler characteristics and phase transitions. The Euler characteristics of a simplicial complex can also be computed using the β via the following formula [56]:

$$\chi = \beta_0 - \beta_1 + \beta_2 - \dots (-1)^{k_{\max}} \beta_{k_{\max}} = \sum_{k=0}^{k_{\max}} (-1)^k \beta_k,$$

where $k_{\max}$ is the maximum dimension that we are computing the cycles.

Furthermore, topological phase transitions are also defined in terms of the β of a simplicial complex [78]. We know that β_0 counts the number of the connected components of a simplicial complex. If we compute the Betti curves as a function of probability in stochastic models, each Betti curve passes through two distinct phases, in a narrow interval: one when it first emerges, and the other when it vanishes [75]. That means that, under similar assumptions as in theoretical models, if the β distribution is unimodal, increasing the density of edges of a brain network will lead to the appearance of β of a higher order. In contrast, smaller Betti numbers will disappear at the vicinity of a topological phase transition.

In Figure 9, we illustrate this property on simplicial complexes obtained from random networks. As the probability increases and so the density of the network is higher, we find a sequence of dominant β_k starting from $k=0$, that change (*i.e.*, k is incremented by one unity) every time a topological phase transition occurs. While the

singularities of the Euler entropy S_χ determine the transitions' location, the crossover of the β characterises which kind of multidimensional hole prevails in each topological phase of the filtration.

Fig 9. Betti number and Euler characteristic approximation. We create a random network for each probability of connection between vertices, and we compute the β and the absolute value of the Euler characteristic. We repeated the experiment 10 times and calculated the mean curves with errors. In this plot, we only show the mean (with errors) of the ten experiments for the Euler and Betti curves. We notice that the absolute value of the Euler characteristic is a good approximation of β .

Curvature:

Curvature is a TDA metric that can link the global network properties described above to local features [64, 79, 80]. When working with brain network data, this will allow us not only to compute topological invariants for the whole-brain set of vertices but also to understand the contribution of specific individual nodal, or subnetwork, geometric proprieties to global properties of the brain network.

Several approaches to defining a curvature for networks are available [19], including some already used in neuroscientific investigations. We will illustrate the curvature approach linked to topological phase transitions, previously introduced for complex systems in [80-82].

To compute the curvature (S3 Code Block), filtration is used to calculate the clique participation rank (*i.e.*, the number of k -cliques in which a vertex i participates for density d) [3], which we denote here by $Cl_{ik}(d)$. The curvature of the vertex based on the participation rank is then defined as:

$$\kappa_i = \sum_{k=1}^{k_{max}} (-1)^{k+1} \frac{Cl_{ik}(d)}{k},$$

where $Cl_{ik} = 1$ since each vertex i participates in a single 1-clique (the vertex itself), and k_{max} the maximum number of vertices that are all-to-all connected in the network.

To link this nodal curvature to the network's global properties, we use the Gauss-Bonnet theorem for networks, through which one can connect a local curvature of a network and its Euler characteristic. Conversely, by summing up all the curvatures of the network across different thresholds, one can reach the alternate sum of the

numbers Cl_k of k -cliques (a subgraph with k all-to-all connected vertices) present in the simplicial complex of the network for a given density threshold $d \in [0, 1]$, according to the following equation:

$$\chi(d) = \sum_{k=1}^N (-1)^{k+1} Cl_k(d).$$

By doing so, we also write the Euler characteristics as a sum of the curvature of all network vertices, *i.e.*,

$$\chi(d) = \sum_{i=1}^N k_i(d).$$

We illustrate the curvature distribution for a functional brain network for densities before and after the transition in Fig 10.

Fig 10. Curvature 3-D plot. Distribution of curvatures in a functional brain network for densities 0.01 (A) and 0.03 (B) after the first topological phase transition. The sum of curvature over all vertices is equal to the Euler characteristic.

Persistent homology:

Homology is a topology branch that investigates objects' shape by studying their holes (or cycles). Persistent homology tracks the emergence of cycles across the evolving simplicial complexes during filtration, allowing us to recognise whether there were homology classes that “persisted” for many filtrations (time here meaning the threshold gap between the birth and death of a cycle) [24, 28]. Importantly, to compute persistent homology, we need to work with a distance matrix, which is the first step in the code below. We can then calculate the simplicial complex's persistence and plot it as a barcode or as a persistence diagram (Fig 3C and E). Here we used the *Gudhi* package for the implementation of those steps [83]. The topological phase transitions

in complex networks [64, 70], which can also be identified between the changes in the dimensionality of the birth/death graphs mentioned above (Fig 3E).

Code example

```
# Converting to appropriate distance matrix (geodesic - strongest edges come
first)
distmat = 1 - matrix

# Computing and plotting barcode
rips_complex = gudhi.RipsComplex(distance_matrix=distmat, max_edge_length=1)
simplex_tree = rips_complex.create_simplex_tree(max_dimension=3)
diag = simplex_tree.persistence()
gudhi.plot_persistence_barcode(diag, legend=True, max_intervals=0)

# Persistence Diagram
gudhi.plot_persistence_diagram(diag, legend=True, max_intervals=0)
```

Discussion

This tutorial has explained some of the main metrics related to two network neuroscience branches - graph theory and TDA -, providing short theoretical backgrounds and code examples accompanied by a publicly available Jupyter Notebook. We innovate by combining hands-on explanations with ready-to-use codes of these subfields and visualisations of simplicial complexes in the brain, thereby hopefully lowering the high threshold necessary for neuroscientists to get acquainted with these new analysis methods, particularly for rsfMRI data. Here, we also innovate by providing realistic visualization of higher-order simplices in brain networks.

Notably, there are limitations and other relevant points that should be kept in mind when working with these metrics. Firstly, it is a common practice in network neuroscience to use null

models for comparison with real data. The idea is to show that the results are different from what one would obtain by chance (or randomly). The generation and comparison with null models must be performed differently for graph theory and TDA, and it is crucial to define what propriety should be kept constant (e.g., the density of the network, or degree distribution). Nevertheless, the computation and discussion of null models are beyond this tutorial's scope and would be an article in itself. A more in-depth discussion of null models in graph theory can be found in [6]. Please see section 4 of [42], and [72] for null models in simplicial complexes (and see Table 1 for further resources).

Moreover, it is crucial to appreciate limitations in interpretation when using these metrics in connectivity-based data. Considering that rsfMRI data is often calculated as a temporal correlation between time series using Pearson's correlation coefficient, a bias on the number of triangles can emerge. For example, suppose areas A and B, and areas C and B are communicating and thus correlated. In that case, a correlation will be present between A and C, even if there would be no actual communication between these vertices [84]. This can affect graph-theoretical metrics such as the clustering coefficient, with networks based on this statistical method being automatically more clustered than random models, and TDA metrics, where the impact depends on how high-order interactions are defined. The proper way to determine and infer high-order interactions in the brain is an ongoing challenge in network neuroscience. Here we simplified our approach using the cliques of a network to define our simplicial complex. For those interested in a more in-depth discussion on the topic, we recommend section 1 and 3 of chapters 7 and 10, respectively, in [6].

The use of weighted matrices can also come with caveats. As mentioned above, various metrics use the sum of weights to compute final nodal values. From that, multiple edges with

low weights might have a final sum equal to few edges with higher weights. How to deal with this limitation and distinguish between these cases is still under discussion. A possible solution was proposed by [85], in which the addition of a tunable parameter in the computation of centralities can allow the researcher to include the number of edges in the total sum, not only the sum of the weights.

Concerning TDA, it is essential to think about limitations in its use due to computational power. The computation of cliques falls in the clique-problem, an NP (nonpolynomial time) problem, thus listing cliques may require exponential time as the size of the cliques or networks grows [69]. For example, if the matrix to be analysed has 60 vertices with a maximum clique size of 23, this will correspond to $\sum \binom{60}{k}$ for $k \in \{0, \dots, 23\}$ cliques, resulting in an enormous amount of time to compute all cliques. What we can do for practical applications is to limit the clique size that can be reached by the algorithm, which determines the dimension of the simplicial complex in which the brain network is represented. This arbitrary constraint implies a theoretical simplification, limiting the space or the dimensionality in which we would analyse brain data. Another issue is that, to finish TDA computations in a realistic timeframe, the researcher might need to establish a maximal threshold/density for convergence even after reducing the maximal clique size. Even though TDA approaches lead to substantial improvements in network science; apart from applications using the Mapper algorithm [86], the limitations mentioned above contribute to losing information on the data's shape [87].

Furthermore, given the early stage of TDA approaches in clinical network neuroscience, it is relevant to recognise that the neurobiological meaning of the metrics mentioned here is still not well defined. Further studies contrasting different neuroscientific techniques with TDA must be done to explain in neurobiological level what a topological metrics represent and how they

correlate with brain functioning. However, it is already possible to use these metrics to differentiate groups [26, 64], and plausible to assume that the interpretation of some classical metrics could be extrapolated to higher orders interactions. For example, the concept of the centralities using pair-wise interactions is used to understand node importance and hubs, the same, in theory, could be applied to the relationships between 3 or more vertices by extending the definition of centrality from networks to simplicial complexes, as done in [88, 89].

Lastly, we would like to briefly mention more general problems in network neuroscience and brain imaging. Before embarking on the application of graph theoretical or topological data analysis, one should be aware of frequent arbitrary decisions such as defining thresholds, using binary or weighted matrices, and controlling for density. Besides, one should think about the differences that arise from using particular atlases and parcellations and their influence on the findings [6, 8, 26, 90-93]. All these factors can impact how credible and reproducible the field of network neuroscience will be, inevitably influencing how appealing the metrics' use might be to clinical practice [91].

Conclusion

Network neuroscience is pivotal in the understanding of brain organisation and function. Graph theory has been the most utilised framework so far, but as the field of network neuroscience expands, newer methods such as TDA are starting to take part in the investigation. To further improve the field, especially in clinical network neuroscience, it is imperative to make the computation of the developed metrics accessible, easy to comprehend, visualize, and efficient. Moreover, researchers must be aware of the crucial decisions that one must make

when executing data analysis and how these can affect studies' results and reproducibility. We hope to have facilitated the comprehension of some aspects of network and topological neuroscience, the computation and visualization of some of its metrics. As a final reminder, we would again suggest that the reader explores our table of resources, and the Jupyter Notebook developed by our team.

References

1. Cowan WM, Harter DH, Kandel ER. The emergence of modern neuroscience: some implications for neurology and psychiatry. *Annu Rev Neurosci*. 2000;23:343-91. doi: 10.1146/annurev.neuro.23.1.343.
2. Rosen BR, Savoy RL. fMRI at 20: has it changed the world? *Neuroimage*. 2012;62(2):1316-24. doi: 10.1016/j.neuroimage.2012.03.004.
3. Sizemore AE, Giusti C, Kahn A, Vettel JM, Betzel RF, Bassett DS. Cliques and cavities in the human connectome. *J Comput Neurosci*. 2018;44(1):115-45. doi: 10.1007/s10827-017-0672-6.
4. Bassett DS, Sporns O. Network neuroscience. *Nature Neuroscience*. 2017;20(3):353. doi: 10.1038/nn.4502.
5. Sporns O. Graph theory methods: applications in brain networks. *Dialogues Clin Neurosci*. 2018;20(2):111-21.
6. Fornito A, Zalesky A, Bullmore E. *Fundamentals of brain network analysis*. 1st ed. San Diego: Academic Press; 2016.
7. Gross JL, Yellen J. *Handbook of graph theory*. 1st ed. Bosa Roca: CRC press; 2003.
8. Bullmore E, Sporns O. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nat Rev Neurosci*. 2009;10(3):186-98. doi: 10.1038/nrn2575.
9. Hart MG, Ypma RJ, Romero-Garcia R, Price SJ, Suckling J. Graph theory analysis of complex brain networks: new concepts in brain mapping applied to neurosurgery. *J Neurosurg*. 2016;124(6):1665-78. doi: 10.3171/2015.4.JNS142683.
10. Raichle ME. The restless brain. *Brain Connect*. 2011;1(1):3-12. doi: 10.1089/brain.2011.0019.
11. Smith SM, Vidaurre D, Beckmann CF, Glasser MF, Jenkinson M, Miller KL, et al. Functional connectomics from resting-state fMRI. *Trends Cogn Sci*. 2013;17(12):666-82. doi: 10.1016/j.tics.2013.09.016.
12. Hallquist MN, Hillary FG. Graph theory approaches to functional network organization in brain disorders: A critique for a brave new small-world. *Netw Neurosci*. 2018;3(1):1-26. doi: 10.1162/netn_a_00054.
13. Bassett DS, Bullmore ET. Small-World Brain Networks Revisited. *Neuroscientist*. 2017;23(5):499-516. doi: 10.1177/1073858416667720.
14. Watts DJ, Strogatz SH. Collective dynamics of 'small-world' networks. *nature*. 1998;393(6684):440. doi: 10.1038/30918.
15. Farahani FV, Karwowski W, Lighthall NR. Application of Graph Theory for Identifying Connectivity Patterns in Human Brain Networks: A Systematic Review. *Front Neurosci*. 2019;13:585. doi: 10.3389/fnins.2019.00585.
16. Zomorodian AJ. *Topology for computing*. 1st ed. New York: Cambridge university press; 2005.

17. Offroy M, Duponchel L. Topological data analysis: A promising big data exploration tool in biology, analytical chemistry and physical chemistry. *Anal Chim Acta*. 2016;910:1-11. doi: 10.1016/j.aca.2015.12.037.
18. Sizemore Blevins A, Bassett DS. Reorderability of node-filtered order complexes. *Physical Review E*. 2020;101(5-1):052311. doi: 10.1103/PhysRevE.101.052311.
19. Blevins AS, Kim JZ, Bassett DSB. Variability in higher order structure of noise added to weighted networks. *arXiv:210103933 [Preprint]*. 2021 [Posted 2021 Jan 11]. Available from: <https://arxiv.org/abs/2101.03933>.
20. Sizemore AE, Phillips-Cremens JE, Ghrist R, Bassett DS. The importance of the whole: Topological data analysis for the network neuroscientist. *Netw Neurosci*. 2019;3(3):656-73. doi: 10.1162/netn_a_00073.
21. Liu TT. Noise contributions to the fMRI signal: An overview. *Neuroimage*. 2016;143:141-51. doi: 10.1016/j.neuroimage.2016.09.008.
22. Greve DN, Brown GG, Mueller BA, Glover G, Liu TT, Function Biomedical Research N. A survey of the sources of noise in fMRI. *Psychometrika*. 2013;78(3):396-416. doi: 10.1007/s11336-012-9294-0.
23. Expert P, Lord LD, Kringelbach ML, Petri G. Editorial: Topological Neuroscience. *Netw Neurosci*. 2019;3(3):653-5. doi: 10.1162/netn_e_00096.
24. Curto C. What can topology tell us about the neural code? *Bulletin of the American Mathematical Society*. 2017;54(1):63-78. doi: 10.1090/bull/1554.
25. Petri G, Expert P, Turkheimer F, Carhart-Harris R, Nutt D, Hellyer PJ, et al. Homological scaffolds of brain functional networks. *Journal of The Royal Society Interface*. 2014;11(101):20140873. doi: 10.1098/rsif.2014.0873.
26. Gracia-Tabuenca Z, Diaz-Patino JC, Arelio I, Alcauter S. Topological Data Analysis Reveals Robust Alterations in the Whole-Brain and Frontal Lobe Functional Connectomes in Attention-Deficit/Hyperactivity Disorder. *eNeuro*. 2020;7(3). doi: 10.1523/ENEURO.0543-19.2020.
27. Centeno EGZ, Santos FN. Notebook for Network and Topological Analysis in Neuroscience. Version v1.0 [software]. 2021 Jan 31. doi: 10.5281/zenodo.4483651
28. Giusti C, Ghrist R, Bassett DS. Two's company, three (or more) is a simplex : Algebraic-topological tools for understanding higher-order structure in neural data. *J Comput Neurosci*. 2016;41(1):1-14. doi: 10.1007/s10827-016-0608-6.
29. Otter N, Porter MA, Tillmann U, Grindrod P, Harrington HA. A roadmap for the computation of persistent homology. *EPJ Data Science*. 2017;6(1):17. doi: 10.1140/epjds/s13688-017-0109-5.
30. Battiston F, Cencetti G, Iacopini I, Latora V, Lucas M, Patania A, et al. Networks beyond pair-wise interactions: Structure and dynamics. *Physics Reports*. 2020;874:1-92. doi: 10.1016/j.physrep.2020.05.004.
31. Giusti C, Pastalkova E, Curto C, Itskov V. Clique topology reveals intrinsic geometric structure in neural correlations. *Proceedings of the National Academy of Sciences, USA*. 2015;112(44):13455-60. doi: 10.1073/pnas.1506407112.
32. Sporns O, Tononi G, Edelman GM. Theoretical neuroanatomy: relating anatomical and functional connectivity in graphs and cortical connection matrices. *Cerebral Cortex*. 2000;10(2):127-41. doi: 10.1093/cercor/10.2.127.
33. Jalili M. Functional Brain Networks: Does the Choice of Dependency Estimator and Binarization Method Matter? *Sci Rep*. 2016;6:29780. doi: 10.1038/srep29780.
34. van den Heuvel MP, de Lange SC, Zalesky A, Seguin C, Yeo BTT, Schmidt R. Proportional thresholding in resting-state fMRI functional connectivity networks and consequences for patient-control connectome studies: Issues and recommendations. *Neuroimage*. 2017;152:437-49. doi: 10.1016/j.neuroimage.2017.02.005.
35. Zhan L, Jenkins LM, Wolfson OE, GadElkarim JJ, Nocito K, Thompson PM, et al. The significance of negative correlations in brain connectivity. *J Comp Neurol*. 2017;525(15):3251-65. doi: 10.1002/cne.24274.
36. Chen G, Chen G, Xie C, Li SJ. Negative functional connectivity and its dependence on the shortest path length of positive network in the resting-state human brain. *Brain Connect*. 2011;1(3):195-206. doi: 10.1089/brain.2011.0025.

37. Murphy K, Birn RM, Handwerker DA, Jones TB, Bandettini PA. The impact of global signal regression on resting state correlations: are anti-correlated networks introduced? *Neuroimage*. 2009;44(3):893-905. doi: 10.1016/j.neuroimage.2008.09.036.
38. Strother SC. Evaluating fMRI preprocessing pipelines. *IEEE Engineering in Medicine and Biology Magazine*. 2006;25(2):27-41. doi: 10.1109/MEMB.2006.1607667.
39. Brown JA, Rudie JD, Bandrowski A, Van Horn JD, Bookheimer SY. The UCLA multimodal connectivity database: a web-based platform for brain connectivity matrix sharing and analysis. *Front Neuroinform*. 2012;6:28. doi: 10.3389/fninf.2012.00028.
40. Biswal BB, Mennes M, Zuo XN, Gohel S, Kelly C, Smith SM, et al. Toward discovery science of human brain function. *Proc Natl Acad Sci U S A*. 2010;107(10):4734-9. doi: 10.1073/pnas.0911855107.
41. Hagberg A, Swart P, S Chult D, editors. Exploring network structure, dynamics, and function using NetworkX. Proceedings of the 7th Python in Science conference (SciPy 2008); 2008 Aug 19-24; Pasadena, USA.
42. Dijkstra EW. A note on two problems in connexion with graphs. *Numerische mathematik*. 1959;1(1):269-71. doi: 10.1007/BF01386390.
43. Golbeck J. Chapter 3 - Network Structure and Measures. In: Golbeck J, editor. *Analyzing the Social Web*. Boston: Morgan Kaufmann; 2013. pp. 25-44.
44. Newman ME. The mathematics of networks. In: Blume L, editor. *The New Palgrave Encyclopedia of Economics*. 2nd ed. Basingstoke: Palgrave Macmillan; 2008. pp. 1-12.
45. Freeman LC. A Set of Measures of Centrality Based on Betweenness. *Sociometry*. 1977;40(1):35-41. doi: 10.2307/3033543.
46. Stam CJ, Tewarie P, Van Dellen E, van Straaten EC, Hillebrand A, Van Mieghem P. The trees and the forest: Characterization of complex brain networks with minimum spanning trees. *Int J Psychophysiol*. 2014;92(3):129-38. doi: 10.1016/j.ijpsycho.2014.04.001.
47. van Dellen E, Sommer IE, Bohlken MM, Tewarie P, Draaisma L, Zalesky A, et al. Minimum spanning tree analysis of the human connectome. *Hum Brain Mapp*. 2018;39(6):2455-71. doi: 10.1002/hbm.24014.
48. Kruskal JB. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*. 1956;7(1):48-50. doi: 10.2307/2033241.
49. Meunier D, Lambiotte R, Bullmore ET. Modular and hierarchically modular organization of brain networks. *Front Neurosci*. 2010;4:200. doi: 10.3389/fnins.2010.00200.
50. Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*. 2008;2008(10):P10008. doi: 10.1088/1742-5468/2008/10/p10008.
51. Carlsson G. Topological methods for data modelling. *Nature Reviews: Physics*. 2020;2(12):697-708. doi: 10.1038/s42254-020-00249-3.
52. Kartun-Giles AP, Bianconi G. Beyond the clustering coefficient: A topological analysis of node neighbourhoods in complex networks. *Chaos, Solitons & Fractals: X*. 2019;1:100004. doi: 10.1016/j.csf.2019.100004.
53. Carlsson G. Topology and data. *Bulletin of the American Mathematical Society*. 2009;46(2):255-308. doi: 10.1090/S0273-0979-09-01249-X.
54. Wang Y, Zhao Y, Nie H, Liu C, Chen J. Disrupted Brain Network Efficiency and Decreased Functional Connectivity in Multi-sensory Modality Regions in Male Patients With Alcohol Use Disorder. *Frontiers in Human Neuroscience*. 2018;12(513). doi: 10.3389/fnhum.2018.00513.
55. Lambiotte R, Rosvall M, Scholtes I. From networks to optimal higher-order models of complex systems. *Nature Physics*. 2019;15(4):313-20. doi: 10.1038/s41567-019-0459-y.
56. Edelsbrunner H, Harer J. *Computational topology: an introduction*. 1st ed. Providence, USA: American Mathematical Society; 2010.
57. Maletić S, Rajković M, Vasiljević D, editors. *Simplicial Complexes of Networks and Their Statistical Properties*. Computational Science – ICCS 2008; 2008 2008//; Berlin, Heidelberg: Springer Berlin Heidelberg.

58. Baudot P. Elements of qualitative cognition: An information topology perspective. *Physics of Life Reviews*. 2019;31:263-75. doi: 10.1016/j.plrev.2019.10.003.

59. Barbarossa S, Sardellitti S. Topological Signal Processing Over Simplicial Complexes. *IEEE Transactions on Signal Processing*. 2020;68:2992-3007. doi: 10.1109/TSP.2020.2981920.

60. Baudot P, Tapia M, Bennequin D, Goillard J-M. Topological Information Data Analysis. *Entropy*. 2019;21(9):869. doi: 10.3390/e21090869.

61. Baudot P, Bennequin D. The Homological Nature of Entropy. *Entropy*. 2015;17(5):3253-318. doi: 10.3390/e17053253.

62. Rosas FE, Mediano PAM, Gastpar M, Jensen HJ. Quantifying high-order interdependencies via multivariate extensions of the mutual information. *Physical Review E*. 2019;100(3):032305. doi: 10.1103/PhysRevE.100.032305.

63. Gatica M, Cofré R, Mediano PAM, Rosas FE, Orio P, Diez I, et al. High-order interdependencies in the aging brain. *BioRxiv* [Preprint]. 2020 [Posted 2020 Mar 18]. Available from: <https://www.biorxiv.org/content/10.1101/2020.03.17.995886v1> doi: 10.1101/2020.03.17.995886.

64. Santos FAN, Raposo EP, Coutinho-Filho MD, Copelli M, Stam CJ, Douw L. Topological phase transitions in functional brain networks. *Physical Review E*. 2019;100(3-1):032414. doi: 10.1103/PhysRevE.100.032414.

65. Inc. PT. Collaborative data science Publisher: Plotly Technologies Inc. Montréal, QC2015.

66. Fan L, Li H, Zhuo J, Zhang Y, Wang J, Chen L, et al. The Human Brainnetome Atlas: A New Brain Atlas Based on Connectional Architecture. *Cerebral Cortex*. 2016;26(8):3508-26. doi: 10.1093/cercor/bhw157.

67. Bakker R, Tiesinga P, Kötter R. The Scalable Brain Atlas: Instant Web-Based Access to Public Brain Atlases and Related Content. *Neuroinformatics*. 2015;13(3):353-66. doi: 10.1007/s12021-014-9258-x.

68. Pardalos PM, Xue J. The maximum clique problem. *Journal of Global Optimization*. 1994;4(3):301-28. doi: 10.1007/BF01098364.

69. Gillis A. The Clique Problem - A Polynomial Time and Non-Heuristic Solution. *viXra:18090188* [Preprint]. 2018 [Posted 2018 Sep 09]. Available from: <https://vixra.org/abs/1809.0188> doi: 10.13140/RG.2.2.14191.07841.

70. Amorim E, Moreira RA, Santos FAN. The Euler Characteristic and Topological Phase Transitions in Complex Systems. *BioRxiv* [Preprint]. 2019 *BioRxiv*:871632 [Posted 2019 Dec 11]. Available from: <http://biorxiv.org/content/early/2019/12/11/871632.abstract> doi: 10.1101/871632.

71. Giri SK, Mellema G. Measuring the topology of reionization with Betti numbers. *arXiv:201212908* [Preprint]. 2020 [Posted 2020 Dec 23]. Available from: <https://arxiv.org/abs/2012.12908>.

72. Lee Y, Lee J, Oh SM, Lee D, Kahng B. Homological percolation transitions in growing simplicial complexes. *arXiv:201012224* [Preprint]. 2021 [Poster 2020 Oct 23, revised 2021 Jan 6]. Available from: <https://arxiv.org/abs/2010.12224>.

73. Santos FAN, da Silva LCB, Coutinho-Filho MD. Topological approach to microcanonical thermodynamics and phase transition of interacting classical spins. *Journal of Statistical Mechanics: Theory and Experiment*. 2017;2017(1):013202. doi: 10.1088/1742-5468/2017/1/013202.

74. Erdős PR. On random graph. *Publicationes Mathematicae*. 1959;6:290-7.

75. Linial N, Peled Y. On the phase transition in random simplicial complexes. *Annals of Mathematics*. 2016;184(3):745-73.

76. Bobrowski O, Skraba P. Homological percolation and the Euler characteristic. *Physical Review E*. 2020;101(3):032304. doi: 10.1103/PhysRevE.101.032304.

77. Speidel L, Harrington HA, Chapman SJ, Porter MA. Topological data analysis of continuum percolation with disks. *Physical Review E*. 2018;98(1):012318. doi: 10.1103/PhysRevE.98.012318.

78. Bobrowski O, Kahle M. Topology of random geometric complexes: a survey. *Journal of Applied and Computational Topology*. 2018;1(3):331-64. doi: 10.1007/s41468-017-0010-0.

79. Weber M, Stelzer J, Saucan E, Naitsat A, Lohmann G, Jost J. Curvature-based Methods for Brain Network Analysis. *arXiv:170700180v2* [Preprint]. 2017 [Posted 2017 Jul 1; revised 2019 May 13]. Available from: <https://ui.adsabs.harvard.edu/abs/2017arXiv170700180W>.

80. Farooq H, Chen Y, Georgiou TT, Tannenbaum A, Lenglet C. Network curvature as a hallmark of brain structural connectivity. *Nat Commun.* 2019;10(1):4937. doi: 10.1038/s41467-019-12915-x.
81. Najman L, Romon P, editors. *Modern approaches to discrete curvature*. 1st ed: Springer; 2017.
82. Wu Z, Menichetti G, Rahmede C, Bianconi G. Emergent complex network geometry. *Sci Rep.* 2015;5:10073. doi: 10.1038/srep10073.
83. Maria C, Boissonnat J-D, Glisse M, Yvinec M. The Gudhi Library: Simplicial Complexes and Persistent Homology. In: Hong H, Yap CK, editors. *Mathematical Software – ICMS 2014*; Seoul, South Korea: Springer-Verlag Berlin Heidelberg; 2014. p. 167-74.
84. Zalesky A, Fornito A, Bullmore E. On the use of correlation as a measure of network connectivity. *Neuroimage.* 2012;60(4):2096-106. doi: 10.1016/j.neuroimage.2012.02.001.
85. Opsahl T, Agneessens F, Skvoretz J. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks.* 2010;32(3):245-51. doi: 10.1016/j.socnet.2010.03.006.
86. Saggat M, Sporns O, Gonzalez-Castillo J, Bandettini PA, Carlsson G, Glover G, et al. Towards a new approach to reveal dynamical organization of the brain using topological data analysis. *Nature Communications.* 2018;9(1):1399. doi: 10.1038/s41467-018-03664-4.
87. Stolz B. *Computational topology in neuroscience*. M.Sc. Thesis, University of Oxford. 2014. Available from: <https://people.maths.ox.ac.uk/porterm/research/Dissertation-stolz2014-Corr.pdf>.
88. Hernández Serrano D, Sánchez Gómez D. Centrality measures in simplicial complexes: Applications of topological data analysis to network science. *Appl Math Comput.* 2020;382:125331. doi: <https://doi.org/10.1016/j.amc.2020.125331>.
89. Estrada E, Ross GJ. Centralities in simplicial complexes. Applications to protein interaction networks. *J Theor Biol.* 2018;438:46-60. doi: <https://doi.org/10.1016/j.jtbi.2017.11.003>.
90. Wang J, Wang L, Zang Y, Yang H, Tang H, Gong Q, et al. Parcellation-dependent small-world brain functional networks: a resting-state fMRI study. *Hum Brain Mapp.* 2009;30(5):1511-23. doi: 10.1002/hbm.20623.
91. Douw L, van Dellen E, Gouw AA, Griffa A, de Haan W, van den Heuvel M, et al. The road ahead in clinical network neuroscience. *Netw Neurosci.* 2019;3(4):969-93. doi: 10.1162/netn_a_00103.
92. Wu Z, Xu D, Potter T, Zhang Y, Alzheimer's Disease Neuroimaging I. Effects of Brain Parcellation on the Characterization of Topological Deterioration in Alzheimer's Disease. *Front Aging Neurosci.* 2019;11:113. doi: 10.3389/fnagi.2019.00113.
93. Eickhoff SB, Yeo BTT, Genon S. Imaging-based parcellations of the human brain. *Nat Rev Neurosci.* 2018;19(11):672-86. doi: 10.1038/s41583-018-0071-7.

Supporting information

S1 Video. 3-D filtration of networks with 2,3,4 and 5-cliques.

S1 Code block. Computation of the Euler characteristic.

S2 Code block. Computation of the Betti numbers.

S3 Code Block. Computation of Curvature.