PLOS ONE

RESEARCH ARTICLE

Metrics for graph comparison: A practitioner's guide

Peter Wills, François G. Meyer *

Department of Applied Mathematics, University of Colorado at Boulder, Boulder, CO, United States of America

* fmeyer@colorado.edu

Citation: Wills P, Meyer FG (2020) Metrics for graph comparison: A practitioner's guide. PLoS ONE 15(2): e0228728. https://doi.org/10.1371/journal.pone.0228728

Editor: Pin-Yu Chen, IBM Research, UNITED STATES

Received: May 17, 2019

Accepted: January 22, 2020

Published: February 12, 2020

Data Availability Statement: Data and software can be obtained here: https://www.github.com/peterewills/netcomp.

Funding: F.G.M was supported by the National Natural Science Foundation (CCF/CIF 1815971, https://www.nsf.gov) and by a French Jean d'Alembert Fellowship. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Comparison of graph structure is a ubiquitous task in data analysis and machine learning, with diverse applications in fields such as neuroscience, cyber security, social network analysis, and bioinformatics, among others. Discovery and comparison of structures such as modular communities, rich clubs, hubs, and trees yield insight into the generative mechanisms and functional properties of the graph. Often, two graphs are compared via a pairwise distance measure, with a small distance indicating structural similarity and vice versa. Common choices include spectral distances and distances based on node affinities. However, there has of yet been no comparative study of the efficacy of these distance measures in discerning between common graph topologies at different structural scales. In this work, we compare commonly used graph metrics and distance measures, and demonstrate their ability to discern between common topological features found in both random graph models and real world networks. We put forward a multi-scale picture of graph structure wherein we study the effect of global and local structures on changes in distance measures. We make recommendations on the applicability of different distance measures to the analysis of empirical graph data based on this multi-scale view. Finally, we introduce the Python library `NetComp` that implements the graph distances used in this work.

1 Introduction

In the era of big data, comparison and matching are ubiquitous tasks. A graph is a particular type of data structure that records the interactions between some collection of agents. These objects are sometimes referred to as "complex networks;" we use the mathematician's term "graph" throughout the paper. This type of data structure relates connections between objects, rather than directly relating the properties of those objects. The interconnectedness of the object in graph data disallows many common statistical techniques used to analyze tabular datasets. The need for new analytical techniques for visualizing, comparing, and understanding graph data has given rise to a rich field of study [1–3].

In this work, we focus on tools for pairwise comparison of graphs. Examples of applications include the two-sample test problem and the change point detection problem. In the former, we compare two populations of graphs using a distance statistic, and we experimentally test

whether both populations could be generated by the same probability distribution. In the latter, we consider a dynamic network formed by a time series of graphs, and the goal is to detect significant changes between adjacent time steps using a distance [4]. Both problems require the ability to effectively compare two graphs. However, the utility of any given comparison method varies with the type of information the user is looking for; one may care primarily about large scale graph features such as community structure or the existence of highly connected "hubs"; or, one may be focused on smaller scale structure such as local connectivity (i.e. the degree of a vertex) or the ubiquity of substructures such as triangles.

Existing surveys of graph distances are limited to observational datasets (e.g., [5] and references therein). While authors try to choose datasets that are exemplars of certain classes of networks (e.g., social, biological, or computer networks), it is difficult to generalize these studies to other datasets.

In this paper, we take a different approach. We consider existing ensembles of random graphs as prototypical examples of certain graph *structures*, which are the building blocks of existing real world networks. We propose therefore to study the ability of various distances to compare two samples randomly drawn from distinct ensembles of graphs. Our investigation is concerned with the relationship between the families of graph ensembles, the structural features characteristic of these ensembles, and the sensitivity of the distances to these characteristic structural features.

The myriad of proposed techniques for graph comparison [6] are severely reduced in number when one requires the practical restriction that the algorithm run in a reasonable amount of time on large graphs. Graph data frequently consists of 10^4 to 10^8 vertices, and so algorithms whose complexity scales quadratically with the size of the graph quickly become unfeasible. In this work, we restrict our attention to approaches where the calculation time scales linearly or near-linearly with the number of vertices in the graph for sparse graphs. We recall that a graph is sparse if the number of edges grows linearly (up to a logarithmic factor) with the number of nodes.

In the past 40 years, many random graph models have been developed that emulate certain features found in real-world graphs [7, 8]. A rigorous probabilistic study of the application of graph distances to these random models is difficult because the models are often defined in terms of a generative process rather than a distribution over the space of possible graphs. As such, researchers often restrict their attention to very small, deterministic graphs (see e.g., [9]) or to very simple random models, such as that proposed by Erdős and Rényi [10]. Even in these simple cases, rigorous probabilistic analysis can be prohibitively difficult. We propose instead a numerical approach where we sample from random graph distributions and observe the empirical performance of various distance measures.

Throughout the work, we examine the observed results through a lens of global versus local graph structure. Examples of global structure include community structure and the existence of well-connected vertices (often referred to as "hubs"). Examples of local structure include the median degree in the graph, or the density of substructures such as triangles. Our results demonstrate that some distances are particularly tuned towards observing global structure, while others naturally observe multiple scales. In both empirical and numerical experiments, we use this multi-scale interpretation to understand why the distances perform the way they do on a given model, or on given empirical graph data.

The paper is structured as follows: in Section 2.2, we introduce the distances used, and establish the state of knowledge regarding each. In Section 2.4, we describe the random graph ensembles that are used to evaluate the various distances. We discuss their structural features, and their respective values as prototypical models for real networks. In Section 2.5 we describe three real networks that we use to further study the performance of the distances. The reader

who is already familiar with the graph models and distances discussed can skip to Section 2.6 to find the description of the contrast statistic that we use to compare graph populations. Experimental results are briefly described in sections 3.1 and 3.2. A detailed discussion is provided in Sections 4.1 and 4.2. Finally, Section **Conclusion** summarizes the work and our recommendations. In Section 7, we introduce and discuss `NetComp`, the Python package that implements the distances used to compare the graphs throughout the paper.

2 Methods

2.1 Notation

We must first introduce the notation used throughout the paper. It is standard wherever possible.

We denote by $G = (V, E, W)$ a graph with vertex set $V = \{1, \ldots, n\}$ and edge set $E \subseteq V \times V$. The function $W: E \longrightarrow \boldsymbol{R}^+$ assigns each edge (i, j) in E a positive weight that we denote $w_{i,j}$. We call $n = |V|$ the **size** of the graph, and denote by $m \stackrel{\text{def}}{=} |E|$ the number of edges. For $i \in V$ and $j \in V$, we say $i \sim j$ if $(i, j) \in E$. The matrix $\boldsymbol{A}$ is called the **adjacency matrix**, and is defined as

$$A_{i,j} \stackrel{\text{def}}{=} \begin{cases} w_{i,j} & \text{if } i \sim j, \\ 0 & \text{otherwise.} \end{cases}$$

The **degree** d_i of a vertex is defined as $d_i \stackrel{\text{def}}{=} \sum_{j \sim i} w_{i,j}$. The **degree matrix** $\boldsymbol{D}$ is the diagonal matrix of degrees, so $D_{i,i} = d_i$ and $D_{i,j} = 0$ for $i \neq j$. The combinatorial **Laplacian matrix** (or just Laplacian) of G is given by $\boldsymbol{L} \stackrel{\text{def}}{=} \boldsymbol{D} - \boldsymbol{A}$. The **normalized Laplacian** is defined as $\mathcal{L} \stackrel{\text{def}}{=} \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}$, where the diagonal matrix $\boldsymbol{D}^{-1/2}$ is given by

$$D_{i,i}^{-1/2} \stackrel{\text{def}}{=} \begin{cases} 1/\sqrt{d_i} & \text{if } d_i \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

We refer to $\boldsymbol{A}$, $\boldsymbol{L}$, and $\mathcal{L}$ as **matrix representations** of G. These are not the only useful matrix representations of a graph, although they are some of the most common. For a more diverse catalog of representations, see [11].

The **spectrum** of a matrix is the sorted sequence of eigenvalues. Whether the sequence is ascending or descending depends on the matrix in question. We denote the k^{th} eigenvalue of the adjacency matrix by $\lambda_k^{\boldsymbol{A}}$, where we order the eigenvalues in descending order

$$\lambda_1^{\boldsymbol{A}} \geq \lambda_2^{\boldsymbol{A}} \geq \ldots \geq \lambda_n^{\boldsymbol{A}}. \tag{1}$$

We denote by $\lambda_k^{\boldsymbol{L}}$ the k^{th} eigenvalue of the Laplacian matrix, and we order these eigenvalues in **ascending** order, so that

$$0 = \lambda_1^{\boldsymbol{L}} \leq \lambda_2^{\boldsymbol{L}} \leq \ldots \leq \lambda_n^{\boldsymbol{L}}. \tag{2}$$

We similarly denote the k^{th} eigenvalue of the normalized Laplacian by $\lambda_k^{\mathcal{L}}$, with

$$0 = \lambda_1^{\mathcal{L}} \leq \lambda_2^{\mathcal{L}} \leq \ldots \leq \lambda_n^{\mathcal{L}}, \tag{3}$$

and we denote by ϕ_k the corresponding eigenvector.

The significance of this convention is that the index k of an eigenvalue λ_k always encodes the frequency of the corresponding eigenvector. To wit, the eigenvector associated with either $\lambda_k^{\boldsymbol{A}}$ or $\lambda_k^{\boldsymbol{L}}$ experiences about k oscillations on the graphs, with $k + 1$ nodal domains [12].

Two graphs G and G' are **isomorphic** if and only if there exists a map between their vertex sets under which the two edge sets are equal; we write $G \cong G'$. If we denote by $\boldsymbol{A}$ and $\boldsymbol{A}'$ the adjacency matrices of G and G' respectively, then $G \cong G'$ if and only if there exists a permutation matrix $\boldsymbol{P}$ such that $\boldsymbol{A}' = \boldsymbol{P}^T\boldsymbol{A}\boldsymbol{P}$.

We say that a distance d requires **node correspondence** when there exist graphs G, G', and H such that $G \cong G'$ but $d(G, H) \neq d(G', H)$. Intuitively, a distance requires node correspondence when one must know some meaningful mapping between the vertex sets of the graphs under comparison.

2.2 Graph distance measures

Let us begin by introducing the distances that we study in this paper, and discussing the state of the knowledge for each. We have chosen both standard and cutting-edge distances, with the requirement that the algorithms be computable in a reasonable amount of time on large, sparse graphs. In practice, this means that the distances must scale linearly or near-linearly in the size in the graph.

We refer to these tools as "distance measures," as many of them do not satisfy the technical requirements of a metric. Although all are symmetric, they may fail one or more of the other requirements of a mathematical metric. This can be very problematic if one hopes to perform rigorous analysis on these distances, but in practice it is not significant. Consider the requirement of identity of indiscernible, in which $d(G, G') = 0$ if and only if $G = G'$. We rarely encounter two graphs where $d(G, G') = 0$; we are more frequently concerned with an approximate form of this statement, in which we wish to deduce that G is similar to G' from the fact that $d(G, G')$ is small.

The distance measures we study divide naturally into two categories, that we now describe. These categories are not exhaustive; many distance measures (including one we employ in the experiments) do not fit neatly into either category. Akoglu et al. [6], whose focus is anomaly detection, provide an alternative taxonomy; our taxonomy refines a particular group of methods they refer to as "feature-based".

2.2.1 Spectral distances. Let us first discuss spectral distances. We briefly review the necessary background; see [11] for a good introduction to spectral methods used in graph comparison.

We first define the adjacency spectral distance; the Laplacian and normalized Laplacian spectral distances are defined similarly. Let G and G' be graphs of size n, with adjacency spectra $\lambda^{\boldsymbol{A}}$ and $\lambda^{\boldsymbol{A}'}$, respectively. The **adjacency spectral distance** between the two graphs is defined as

$$d_{\boldsymbol{A}}(G, G') \stackrel{\text{def}}{=} \sqrt{\sum_{i=1}^{n} (\lambda_i^{\boldsymbol{A}} - \lambda_i^{\boldsymbol{A}'})^2},$$

which is just the distance between the two spectra in the ℓ_2 metric. We could use any ℓ_p metric here, for $p \in [0, \infty]$. The choice of p is informed by how much one wishes to emphasize outliers; in the limiting case of $p = 0$, the metric returns the measure of the set over that the two vectors are different, and when $p = \infty$ only the largest element-wise difference between the two vectors is returned. Note that for $p < 1$ the ℓ_p distances are not true metrics (in particular, they fail the triangle inequality) but they still may provide valuable information. For a more detailed discussion on ℓ_p norms, see [13].

The Laplacian and normalized Laplacian spectral distances d_L and $d_{\mathcal{L}}$ are defined in the exact same way. In general, one can define a spectral distance for any matrix representation of

a graph; for results on more than just the three we analyze here, see [11]. We note that spectral distances do not require node correspondence.

An important property of the normalized Laplacian spectral distance is that it can be used to compare graphs of different sizes (see e.g., [14]).

In practice, it is often the case that only the first k eigenvalues are compared, where $k \ll n$. We still refer to such truncated spectral distances as spectral distances. When using spectral distances, it is important to keep in mind that the adjacency spectral distance compares the *largest k* eigenvalues, whereas the Laplacian spectral distances compare the *smallest k* eigenvalues. Comparison using the first k eigenvalues $\lambda_k^{\boldsymbol{A}}$ for small k allows one to focus on the community structure of the graph, while ignoring the local structure of the graph [15]. Inclusion of the highest-k eigenvalues $\lambda_k^{\boldsymbol{A}}$ allows one to discern local features as well as global. This flexibility allows the user to target the particular scale at which she wishes to study the graph, and is a significant advantage of the spectral distances.

The three spectral distances used here are not true metrics. This is because there exist graphs G and G' that are co-spectral but not isomorphic. That is to say, adjacency cospectrality occurs when $\lambda_i^{\boldsymbol{A}} = \lambda_i^{\boldsymbol{A}'}$ for all $i = 1, \ldots, n$, so $d_{\boldsymbol{A}}(G, G') = 0$, but $G \ncong G'$. Similar notions of cospectrality exist for all matrix representations; graphs that are co-spectral with respect to one matrix representation are not necessarily co-spectral with respect to other representations.

Little is known about cospectrality, save for some computational results on small graphs [16] and trees [11]. Schwenk proved that a sufficiently large tree nearly always has a co-spectral counterpart [17]. This result was extended recently to include a wide variety of random trees [18]. However, results such as these are not of great import to us; the graphs examined are large enough that we do not encounter cospectrality in our numerical experiments. A more troubling failure mode of the spectral distances would be when the distance between two graphs is very small, but the two graphs have important topological distinctions. In Section **Discussion**, we provide further insight into the effect of topological changes on the spectra of some of the random graph models we study.

The consideration above addresses the question of how local changes affect the overall spectral properties of a graph. Some limited computational studies have been done in this direction. For example, Farkas et al. [19] study the transition of the adjacency spectrum of a small world graph as the disorder parameter increases. As one might expect, the authors in [19] observe the spectral density transition from a highly discontinuous density (which occurs when the disorder is zero and the graph is a ring-like lattice) to Wigner's famous semi-circular shape [20] (which occurs when the disorder is maximized, so that the graph is roughly equivalent to an uncorrelated random graph).

From an analytical standpoint, certain results in random matrix theory inform our understanding of fluctuations of eigenvalues of the uncorrelated random graph (see Section **Random Graph Models** for a definition). These results hold asymptotically as we consider the k^{th} eigenvalue of a graph of size n, where $k = \alpha n$ for $\alpha \in (0, 1]$. In this case, O'Rourke [21] has shown that the the eigenvalue λ_k is asymptotically normal with asymptotic variance $\sigma^2(\lambda_k) = C(\alpha) \log n/n$. An expression for the constant $C(\alpha)$ is provided; see Remark 8 in [21] for the detailed statement of the theorem. This result can provide a heuristic for spectral fluctuations in some random graphs, but when the structure of these graphs diverges significantly from that of the uncorrelated random graph, then results such as these become less informative.

Another common question is that of interpretation of the spectrum of a given matrix representation of a graph. How are we to understand the shape of the empirical distribution of eigenvalues? Specifically, one might study the overall shape of the spectral density, or the value of individual eigenvalues separated from the bulk. Can we interpret the eigenvalues which

separate from this bulk in a meaningful way? The answer to this question depends, of course, on the matrix representation in question. Let us focus first on the Laplacian matrix $\boldsymbol{L}$, the interpretation of that is the clearest.

The first eigenvalue of $\boldsymbol{L}$ is always $\lambda_1^{\boldsymbol{L}} = 0$, with the eigenvector being the vector of all ones, $\mathbf{1} \in \mathbb{R}^n$. It is a well-known result that the multiplicity of the zero eigenvalue is the number of connected components of the graph, i.e. if $0 = \lambda_k^{\boldsymbol{L}} < \lambda_{k+1}^{\boldsymbol{L}}$, then there are precisely k connected components of the graph [22]. Furthermore, in such a case, the first k eigenvectors can be chosen to be the indicator functions of the components. There exists a relaxed version of this result: if the first k eigenvalues are very small (in a sense properly defined), then the graph can be strongly partitioned into k clusters (see [15] for the rigorous formulation of the result). This result justifies the use of the Laplacian in spectral clustering algorithms, and can help us understand the interplay between the presence of small eigenvalues and the presence of communities in the ensembles of random graphs studied in Section 3.1.1.

The eigenvalues of the Laplacian can be interpreted as vibrational frequencies in a manner similar to the eigenvalues of the continuous Laplacian operator ∇^2. To understand this analogy, consider the graph as embedded in a plane, with each vertex representing an oscillator of mass one and each edge a spring with elasticity one. Then, for small oscillations perpendicular to the plane, the Laplacian matrix is precisely the coupling matrix for this system, and the eigenvalues give the square of the normal mode frequencies, $\omega_i = \sqrt{\lambda_i^{\boldsymbol{L}}}$. For a more thorough discussion of this interpretation of the Laplacian, see [23].

Maas [24] suggests a similar interpretation of the spectrum of the adjacency matrix $\boldsymbol{A}$. Consider the graph as a network of oscillators, embedded in a plane as previously discussed. Additionally, suppose that each vertex is connected to so many external non-moving points (by edges with elasticity one) so that the graph becomes regular with degree d. The frequencies of the normal modes of this structure then connect to the eigenvalues of $\boldsymbol{A}$ via $\omega_i = \sqrt{d - \lambda_i^{\boldsymbol{A}}}$. If the graph is already regular with degree d, then this interpretation is consistent with the previous, since the eigenvalues of $\boldsymbol{L} = d\boldsymbol{I} - \boldsymbol{A}$ are just $\lambda_i^{\boldsymbol{L}} = d - \lambda_i^{\boldsymbol{A}}$.

2.2.2 Matrix distances. The second class of distances we discuss are called *matrix distances*, and consist of direct comparison of the structure of pairwise affinities between vertices in a graph (see [9] for a detailed discussion on matrix distances). These affinities are frequently organized into matrices, and the matrices can then be compared, often via an entry-wise ℓ_p norm. Matrix distances all require node correspondence.

We have discussed spectral methods for measuring distances *between* two graphs; to introduce the matrix distances, we begin by focusing on methods for measuring distances *on a graph*; that is to say, the distance $\delta(v, w)$ between two vertices $v, w \in V$. Just a few examples of such distances include the shortest path distance [25], the effective graph resistance [26], and variations on random-walk distances [27]. Of those listed above, the shortest path distance is the oldest and the most thoroughly studied; in fact, it is so ubiquitous that "graph distance" is frequently used synonymously with shortest path distance [28].

There are important differences between the distances δ that we might choose. The shortest path distance considers only a single path between two vertices. In comparison, the effective graph resistance takes into account all possible paths between the vertices, and so measures not only the length, but the *robustness* of the communication between the vertices.

How do these distances *on a graph* help us compute distances *between graphs*? Let us denote by $\delta : V \times V \to \mathbb{R}$ a generic distance on a graph. We need assume very little about this function, besides it being real-valued; in particular, it need not be symmetric, and we can even allow $\delta(v, v) \neq 0$. When we say "distance" we implicitly assume that smaller values imply

greater similarity; however, we can also carry out this approach with a "similarity score", in which larger values imply greater similarity. Recalling that the vertices $v \in V = \{1, \ldots, n\}$ are labelled with natural numbers, we can then construct a matrix of pairwise distances $\boldsymbol{M}$ via $M_{i,j} \stackrel{\text{def}}{=} \delta(i, j)$. The idea behind what we refer to as **matrix distances** is that this matrix $\boldsymbol{M}$ carries important structural information about the graph.

Consider two graphs $G = (V, E)$ and $G' = (V, E')$ defined on the same vertex set. Given a graph distance $\delta(\cdot, \cdot)$, let $\boldsymbol{M}$ and $\boldsymbol{M}'$ be the matrices of pairwise distances between vertices in the graph G and G' respectively. We define the distance d induced by δ between G and G' as follows,

$$d(G, G') \stackrel{\text{def}}{=} \| \boldsymbol{M} - \boldsymbol{M}' \|, \tag{4}$$

where $\| \cdot \|$ is a norm we are free to choose. In principle, we could use metrics, or even similarity functions here, at the risk of the function d losing some desirable properties.

Let us elucidate a specific example of such a distance; in particular, we show how the edit distance conforms to this description. Let $\delta(v, w)$ be defined as

$$\delta(v, w) = \begin{cases} 1 & \text{if } v \sim w, \\ 0 & \text{else.} \end{cases} \tag{5}$$

Then the matrix $\boldsymbol{M}$ is just the adjacency matrix $\boldsymbol{A}$. If we use the norm

$$\| \boldsymbol{M} \| = \sum_{i,j=1}^{n} |M_{i,j}|, \tag{6}$$

then we call the resulting distance $d(G, G') \stackrel{\text{def}}{=} \| \boldsymbol{A} - \boldsymbol{A}' \|$ the **edit distance**.

Of course, the usefulness of such a distance is directly dependent on how well the matrix $\boldsymbol{M}$ reflects the topological structure of the graph. The edit distance focuses by definition on local structure; it can only see changes at the level of edge perturbations. If significant volume changes are happening in the graph, then the edit distance detects these changes, as do other matrix distances.

To compensate for such trivial first order changes (changes in volume) we match the expected volume of the models under comparison (see Section 3.1). We can then study whether distances can detect structural changes.

We also implement the resistance-perturbation distance, first discussed in [9]. This distance takes the effective graph resistance $R(u, v)$, defined in [26], as the measure of vertex affinity. This results in a (symmetric) matrix of pairwise resistances $\boldsymbol{R}$. The resistance-perturbation distance (or just resistance distance) is based on comparing these two matrices in the entry-wise ℓ_1 norm given in Eq (6).

Unlike the edit distance, the resistance distance is designed to detect changes in connectivity between graphs. A recent work [29] discusses the efficacy of the resistance distance in detecting community changes.

Finally, we study DeltaeCon, a distance based on the fast belief propagation method of measuring node affinities [30]. To compare graphs, this method uses the fast belief propagation matrix

$$\mathbf{S} \stackrel{\text{def}}{=} [\boldsymbol{I} + \epsilon^2 \boldsymbol{D} - \epsilon \boldsymbol{A}]^{-1}, \tag{7}$$

and compares the two representations **S** and **S**′ via the Matusita difference,

$$d(G, G') = \sqrt{\sum_{i,j} \left(\sqrt{S_{i,j}} - \sqrt{S'_{i,j}} \right)^2}. \tag{8}$$

Note that the matrix **S** can be rewritten in a matrix power series as

$$\mathbf{S} \approx \boldsymbol{I} + \epsilon \boldsymbol{A} + \epsilon^2 (\boldsymbol{A}^2 - \boldsymbol{D}) + \dots \tag{9}$$

and so takes into account the influence of neighboring vertices in a weighted manner, where neighbors separated by paths of length k have weight ϵ^k. Fast belief propagation is designed to model the diffusion of information throughout a graph [31], and so should in theory be able to perceive both global and local structures. Although empirical tests are performed in [30], no direct comparison to other modern methods is presented.

2.2.3 Feature-based distances. These two categories do not cover all possible methods of graph comparison. The computer science literature explores various other methods (e.g., see [6], Section 3.2 for a comprehensive review), and other disciplines that apply graph-based techniques often have their own idiosyncratic methods for comparing graphs extracted from data.

One possible method for comparing graphs is to look at specific "features" of the graph, such as the degree distribution, betweenness centrality distribution, diameter, number of triangles, number of k-cliques, etc. For graph features that are vector-valued (such as degree distribution) one might also consider the vector as an empirical distribution and take as graph features the sample moments (or quantiles, or statistical properties). A *feature-based distance* is a distance that uses comparison of such features to compare graphs.

Of course, in a general sense, all methods discussed so far are feature based; however, in the special case where the features occur as values over the space $V \times V$ of possible node pairings, we choose to refer to them more specifically as *matrix distances*. Similarly, if the feature in question is the spectrum of a particular matrix realization of the graph, we call the method a spectral distance.

In [32], a feature-based distance called NetSimile is proposed, which focuses on local and egonet-based features (e.g., degree, volume of egonet as fraction of maximum possible volume, etc.). If we are using k features, the method aggregates a feature-vertex matrix of size $k \times n$. This feature matrix is then reduced to a "signature vector" (a process the authors in [32] call "aggregation") that consists of the mean, median, standard deviation, skewness, and kurtosis of each feature. These signature vectors are then compared in order to obtain a measure of distance between graphs.

In the neuroscience literature in particular, feature-based methods for comparing graphs are popular [33, 34]. In [35], the authors use graph features such as modularity, shortest path distance, clustering coefficient, and global efficiency to compare functional connectivity networks of patients with and without schizophrenia. Statistics of these features for the control and experiment groups are aggregated and compared using standard statistical techniques.

We implement NetSimile as a prototypical feature-based method. It is worth noting that the general approach could be extended in almost any direction; any number of features could be used (which could take on scalar, vector, or matrix values) and the aggregation step can include or omit any number of summary statistics on the features, or can be omitted entirely. We implement the method as it is originally proposed, with the caveat that calculation of many of these features is not appropriate for large graphs, as they cannot be computed in linear or near-linear time. A scalable modification of NetSimile would utilize features that can be calculated (at least approximately) in linear or near-linear time.

2.2.4 Learning graph kernels. Given the diversity of structural features in graphs, and the difficulty of designing by hand the set of features that optimizes the graph embedding, several researchers have proposed recently to learn the embedding from massive datasets of existing networks. Such algorithms learn an embedding [36] from a set of graphs into Euclidean space, and then compute a notion of similarity between the embedded graphs (e.g., [37–39] and references therein). The metric that is learnt can be tailored to a specific application (e.g., [39–45]).

All these approaches rely on the extension of convolutional neural networks to non Euclidean structures, such as manifolds and graphs (e.g., [46–49] and references therein). The core scientific question becomes: how does one implement the convolution units that are in the network? Two methods have been proposed. The first method performs the convolution in the spectral domain [50], (defined by the eigenspace of the graph Laplacian). These data-dependent convolutions can be performed directly in the spatial domain (using polynomials of the Laplacian [51]) or in the spectral domain (in the eigenspace of the Laplacian). Purely "in-the-graph" methods have also been proposed where the convolution is implemented using an aggregation process (e.g., [42, 52, 53] and references therein).

Graph kernels [54] are typically not injective (two graphs can be perfectly similar without being the same), and rarely satisfies the triangular inequality. There have been some recent attempts at identifying the classes of kernels that are injective [55, 56]. The question can be rephrased in terms of how expressive is the embedding from the space of graphs to Euclidean space, i.e. how often do two distinct graphs are mapped to same point [57]. The authors in [55, 56] have proved that graph neural networks are as expressive as the Weisfeiler-Lehman graph isomorphism test: if two graphs are mapped to distinct points by the embedding, then the Weisfeiler-Lehman graph test would consider these graphs to be distinct (non isomorphic).

2.2.5 Comparing graphs of different sizes. The distance measures described in the previous paragraphs are defined for two graphs that have the same size. In practice, one often needs to compare graphs of different sizes. Inspired by the rich connections between graph theory and geometry, one can define a notion of distance between any two graphs by extending the notion of distance between metric spaces [58]. The construction proceeds as follows: each graph is represented as a metric space, wherein the metric is simply the shortest distance on the graph. Two graphs are equivalent if there exists an isomorphism between the graph—represented as metric spaces. Finally, one can define a distance between two graphs G_1 and G_2 (or rather between the two classes of graph isometric to G_1 and G_2 respectively) by considering standard notions of distances between isometry classes of metric spaces [59]. Examples of such distances include the Gromov-Hausdorff distance [59], the Kantorovich-Rubinstein distance and the Wasserstein distance [60], which both require that the metric spaces be equiped with probability measures. The Gromov-Hausdorff distance computes the infimum of the Hausdorff distance between the isometric embeddings of two metric spaces into a common one. In plain English, this distance measures the residual error after trying to "optimally align" two metric spaces using deformations of these spaces that preserve distances (isometries). Because the search for the optimal alignment (embedding) is over such a vast space of functions, the Gromov-Hausdorff does not lend itself to practical applications (but see [61]).

On the other hand, the Wasserstein-Kantorovich-Rubinstein distance, also known as the "Earth Mover's distance" in the engineering literature, has been used extensively in probability and pattern recognition (e.g., [62–64] and references therein). The Wasserstein distance can be interpreted as the cost of transporting a measure from one metric space to a second measure defined on a second metric space; the cost increases with the distance between the metric spaces and the proportion of the measure that needs to be transported. These concepts have just recently been applied to the case of measuring distances between graphs. Given a graph G,

one can associate a measure on the graph (e.g., defined by a histogram of the degrees [65, 66], a Gaussian measure with a covariance matrix given by the pseudo-inverse of the graph Laplacian [67], or a uniform measure on the graph [68]), and a notion of cost between nodes (e.g., the Bures distance [67], the shortest distance between two nodes [68] assuming the node correspondence between the graphs has been established).

The computational complexity of the estimation of the Wasserstein distance remains prohibitively high for large graph: the cost is $mn^2 + m^2n$, where m is the number of edges, and n is the number of nodes. A closed form expression of the Wasserstein distance can be derived when the measure on each graph is a Gaussian measure [67]. In this case the Wasserstein distance is the Bures distance between their respective covariance matrices. This computation is further simplified when the covariance matrices are diagonal, since the Bures distance becomes then the Hellinger distance (e.g., [69, 70] and references therein). The rich connection between distances between metric spaces, optimal transport, and metrics on the cone of positive semidefinite matrices is clearly beyond the scope of the current study; it will certainly provide interesting avenues for future studies.

The relevance of the current study to this burgeoning research area stems from the exploration of the relationship between the structural features characteristic of several graph ensembles and the sensitivity of the distances to these features. The distributions associated with these features can then be used to define a probability measure associated with a given graph (e.g., [71] where the distribution of hitting times is used to characterize a functional brain connectivity network).

2.3 Computational efficiency

2.3.1 Algorithmic complexity. In many interesting graph analysis scenarios, the sizes of the graphs to be analyzed are on the order of millions or even billions of vertices. For example, the social network defined by Facebook users has over 2.3 billion vertices as of 2018. In scenarios such as these, any algorithm of complexity $\mathcal{O}(n^2)$ becomes unfeasible; although in principle it is possible that the constant hidden in $\mathcal{O}()$ would be so small it would make up for the n^2 term in the complexity, in practice this is not the case. This motivates the requirement that algorithms be of near-linear complexity. When the complexity of the distance depends on the graph volume m, we assume that the graph is sparse and m is a linear function (up to a logarithmic factor) of the size n.

This challenge motivates the previously stated requirement that all algorithms be of linear or near-linear complexity. We say an algorithm is **linear** if it is $\mathcal{O}(n)$; it is **near-linear** if it is $\mathcal{O}(n \log a_n)$ where a_n is asymptotically bounded by a polynomial. We use the notation $a_n = \mathcal{O}(b_n)$ in the standard way; for a more thorough discussion of algorithmic complexity, including definitions of the Landau notations, see [72].

Table 1 displays the algorithmic complexity of each distance measure we compare. We assume that factors such as graph weights and quality of approximation are held constant, leading to simpler expressions here than appear in cited references. Spectral distances have equivalent complexity, since they all all amount to performing an eigendecomposition on a symmetric real matrix. For DeltaCon and the resistance distance, there are approximate algorithms as well as exact algorithms; we list the complexity of both. Although we use the exact versions in the experiments, in practice the approximate version would likely be used if the graphs to be compared are large.

Of particular interest are the highly parallelizable randomized algorithms which can allow for extremely efficient matrix decomposition. In [73], the authors review many such algorithms, and discuss in particular their applicability to determining principal eigenvalues. The

Table 1. Distance measures and complexity. The size (of the larger) graph is n; the number of edges is m. For the spectral decomposition, k denotes the number of principal eigenvalues we wish to find.

Distance Measure	Complexity	Ref.
Edit Distance	$\mathcal{O}(m)$	[74]
Resistance Distance (Exact)	$\mathcal{O}(n^2)$	[9]
Resistance Distance (Approximate)	$\mathcal{O}(m)$	[9]
DeltaCon (Exact)	$\mathcal{O}(n^2)$	[30]
DeltaCon (Approximate)	$\mathcal{O}(m)$	[30]
NetSimile	$\mathcal{O}(n \log n)$	[32]
Spectral Distance	$\mathcal{O}(nk^2)$	[73]

https://doi.org/10.1371/journal.pone.0228728.t001

computation complexity in Table 1 for the spectral distances is based on their simplified analysis of the Krylov subspace methods, that states that the approach is $\mathcal{O}(kT_{\text{mult}} + (m+n)k^2)$, where T_{mult} is the cost of matrix-vector multiplication for the input matrix. Since the input matrices are sparse, $T_{\text{mult}} = \mathcal{O}(n)$, and $m + n = \mathcal{O}(n)$. Although the eigensolver uses the implicitly restarted Arnoldi method, if implementing such a decomposition on large matrices, the use of a randomized algorithm could lead to a significant increase in efficiency.

2.3.2 Comparison of runtimes on graphs on small graphs. In Section 3.1, we perform the experiments on small graphs, consisting of only 1,000 nodes.

In application, the graphs under comparison can vary from hundreds up to billions of nodes. We focus on smaller graphs primarily so that the computation of the distances is tractable even on a small personal computer.

Of course, the time it takes to calculate a given distance depends highly on the implementation of that distance. The runtimes reported below use the implementations in NetComp [75]. These implementations are not highly optimized; spectral calculations depends on the standard spectral solvers that come with `scipy`, a standard computational package in Python. These leverage sparse data structures when available.

For the resistance distance and DeltaCon, the distance has both an exact form which has $\mathcal{O}(n^2)$ complexity, and an approximate form which has $\mathcal{O}(m)$ complexity. We use the exact forms in our calculations, and these are the forms implemented in NetComp [75]. For DeltaCon, the approximate form is implemented in MATLAB, and the code is available on the author's website, http://web.eecs.umich.edu/~dkoutra/. For the resistance distance, the authors of [9] have released an implementation of the approximate resistance distance in MATLAB, which can be found on GitHub at https://github.com/natemonnig/Resistance-Perturbation-Distance. We hope to include Python implementations of these fast approximate distances in NetComp in the near future.

Table 2 shows the results of our runtime experiments. We compare mean and standard deviations of runtimes for the various distances.

These are computed on small graphs, of size n = 100, 300, and 1, 000.

As one might expect, the edit distance is by far the most efficient, as it is simply a difference between and summation over two sparse matrices. NetSimile is notably slow in our experiments. This is due to inefficient implementation—most of the work of calculating the various metrics used by NetSimile is done by leveraging NetworkX, a common network analysis library in Python. Although NetworkX is very simple and clear to work with, it is not designed for maximal efficiency or scalability, as is evidenced by the above experiments.

We believe it is valuable for the user to get a rough estimate of the efficiency of the easily-available implementations of the distances discussed in this work. However, much more

Table 2. Runtimes for distance various distance measures, for graphs of size $n = 100$ and $n = 300$. Each distance is calculated $N = 500$ times. Each sample generates two Erdős-Rényi random graphs with parameter $p = 0.15$, and times the calculation of the distance between the two graphs. All distances are implemented in the `NetComp` library, which can be found on GitHub at [75].

Distance Measure	Computational Time ($n = 100$)
Edit Distance	$8.2 \times 10^{-5} \pm 4.5 \times 10^{-5}$
DeltaCon	$3.1 \times 10^{-3} \pm 7.4 \times 10^{-4}$
Resistance Dist.	$7.5 \times 10^{-3} \pm 1.4 \times 10^{-3}$
Spectral (Adjacency)	$1.1 \times 10^{-2} \pm 1.1 \times 10^{-3}$
Spectral (Laplacian)	$1.2 \times 10^{-2} \pm 4.7 \times 10^{-3}$
Spectral (Normalized Laplacian)	$1.5 \times 10^{-2} \pm 9.4 \times 10^{-4}$
NetSimile	$2.3 \times 10^{-1} \pm 6.3 \times 10^{-2}$
Distance Measure	**Computational Time ($n = 300$)**
Edit Distance	$5.7 \times 10^{-4} \pm 9.9 \times 10^{-4}$
DeltaCon	$1.4 \times 10^{-2} \pm 6.6 \times 10^{-3}$
Resistance Dist.	$8.8 \times 10^{-2} \pm 5.4 \times 10^{-2}$
Spectral (Adjacency)	$1.5 \times 10^{-1} \pm 8.9 \times 10^{-3}$
Spectral (Laplacian)	$1.5 \times 10^{-1} \pm 9.8 \times 10^{-3}$
Spectral (Normalized Laplacian)	$1.6 \times 10^{-1} \pm 6.6 \times 10^{-3}$
NetSimile	$5.5 \times 10^{-1} \pm 1.1 \times 10^{-2}$
Distance Measure	**Computational Time ($n = 1,000$)**
Edit Distance	$4.2 \times 10^{-3} \pm 1.5 \times 10^{-3}$
DeltaCon	$8.8 \times 10^{-2} \pm 6.4 \times 10^{-3}$
Resistance Dist.	$5.9 \times 10^{-1} \pm 4.3 \times 10^{-2}$
Spectral (Adjacency)	$1.3 \pm 5.5 \times 10^{-1}$
Spectral (Laplacian)	$1.3 \pm 1.6 \times 10^{-1}$
Spectral (Normalized Laplacian)	$1.4 \pm 3.7 \times 10^{-1}$
NetSimile	$2.5 \pm 1.8 \times 10^{-1}$

https://doi.org/10.1371/journal.pone.0228728.t002

efficient implementations are possible for each given distance; these implementations must be carefully designed to be optimal for the particular use-case. A thorough empirical comparison of the runtimes of optimized implementations of each of these distances would be very illuminating, but would require considerable care in order to be done equitably, and is well beyond the scope of this work.

2.4 Random graph models

Random graph models have long been used as a method for understanding topological properties of graph data that occurs in the world. The uncorrelated random graph model of Erdős and Rényi [10] is the simplest model, and provides a null model akin to white noise. This probabilistic model has been analysed thoroughly [76]. Unfortunately, the uniform topology of the model does not accurately model empirical graph data. The stochastic blockmodel is an extension of the uncorrelated random graph, but with explicit community structure reflected in the distribution of edge density.

Models such as preferential attachment [7] and the Watts-Strogatz model [8] have been designed to mimic properties of observed graphs. Very little can be said about these models analytically, and thus much of what is understood about them is computational. The two-dimensional square lattice is a quintessential example of a highly structured and regular graph.

Finally, we restrict the present study to unlabelled and undirected graphs, with no self-loops. Although directed graphs are of great practical importance [77], the mathematical analysis of directed graphs is far more complex.

Most of the models in this work are sampled via the Python package NetworkX [78]; details of implementation can be found in the source code of the same. Some of the models we use are most clearly defined via their associated probability distribution, while others are best described by a generative mechanism. We introduce the models roughly in order of complexity.

2.4.1 The uncorrelated random graph. The **uncorrelated** Erdős-Rényi **random graph** is a random graph in which each edge exists with probability p, independent of the existence of all others. We denote this distribution of graphs by $G(n, p)$. The spectral density of the $\lambda^{\boldsymbol{A}}$ forms a semi-circular shape, first described by Wigner [20], of radius $\sqrt{np(1-p)}$, albeit with a single eigenvalue $\lambda_1^{\boldsymbol{A}} \approx np$ separate from the semicircular bulk [19].

We employ the uncorrelated random graph as the null model in many experiments. It is, in some sense, a "structureless" model; more specifically, the statistical properties of each edge and vertex in the graph are exactly the same. This model fails to produce many of properties observed in empirical networks, that motivates the use of alternative graph models.

2.4.2 The stochastic blockmodel. One important property of real world networks is community structure. Vertices often form densely connected communities, with the connection between communities being sparse, or non-existent. This motivates the use of the **stochastic blockmodel**. In this model, the vertex set can be partitioned into two non-overlapping sets C_1 and C_2 referred to as "communities",

$$V = C_1 \cup C_2. \tag{10}$$

Each edge $e = (i, j)$ exists independently with probability p if i and j are in the same community, and q if i and j are in distinct communities. In this work, we use "balanced" communities, whose sizes are equal (up to one vertex in either direction).

The stochastic blockmodel is a prime example of a model that exhibits global structure without any meaningful local structure. In this case, the global structure is the partitioned nature of the graph as a whole. On a fine scale, the graph looks like an uncorrelated random graph. We use the model to determine which distances are most effective at discerning global (and in particular, community) structure.

The stochastic blockmodel is at the cutting edge of rigorous probabilistic analysis of random graphs. Abbe et al. [79] have recently proven a strict bound on community recovery, showing in exactly what regimes of p and q it is possible to detect the communities, and assign the correct label to each node.

Generalizations of this model exist in which there are K communities of arbitrary size. Furthermore, each community need not have the same parameter p, and each community *pair* need not have the same parameter q.

2.4.3 Preferential attachment models. Another often-studied feature of real world networks is the degree distribution. In practice, the distribution is estimated using a histogram.

The degree distribution of an uncorrelated random graph is binomial, and so it has tails that decay exponentially for large graphs (as $n \to \infty$). However, in real world graphs such as computer networks, human neural nets, and social networks, the measured degree distribution has a power-law tail [7], $\mathbb{P}(d) \propto d^{-\gamma}$ where $\gamma \in [2, 3]$. Such distributions are often also referred to as "scale-free".

The **preferential attachment model** is a scale-free random graph model. Although first described by Yule in 1925 [80], the model did not achieve its current popularity until the work of Barabási and Albert in 1999 [7].

The model has two parameters, l and n. The latter is the size of the graph, and the former controls the density of the graph. We require that $1 \leq l < n$. The generative procedure for sampling from this distribution proceeds as follows. Begin by initializing a star graph with $l + 1$ vertices, with vertex $l + 1$ having degree l and all others having degree 1. Then, for each $l + 1 < i \leq n$, add a vertex, and randomly attach it to l vertices already present in the graph, where the probability of i attaching to v is proportional to to the degree of v. We stop once the graph contains n vertices.

The constructive description of the algorithm does not yield itself to simple analysis, and so less is known analytically about the preferential attachment model than the uncorrelated random graph or the stochastic blockmodel (but see [81, 82] for some basic properties of this model). There are few results about the spectrum of the various matrices. In [83], the authors prove that if $\lambda_1^A \geq \ldots \lambda_k^A$ are the k largest eigenvalues of the adjacency matrix, and if $d_1 \geq \ldots \geq d_k$ are the k largest degrees, then

$$\lambda_i^A = (1 + \mathcal{O}(1))\sqrt{d_i} \quad \text{with high probability.} \tag{11}$$

These results are proven on a model with a slightly different generative procedure; we do not find that they yield a particularly good approximation for our experiments that are conducted at the quite low $n = 100$. In [19], the authors demonstrate numerically that the adjacency spectrum exhibits a triangular peak with power-law tails.

Having a high degree makes a vertex more likely to attract more connections, so the graph quickly develops strongly connected "hubs," or vertices with very high degree, which cannot be found in the Erdős-Rényi model. This impacts both the global and local structure of the graph. Hubs are by definition global structures, as they touch a significant portion of the rest of the graph, making path lengths shorter and increasing connectivity throughout the graph. On the local scale, vertices in the graph tend to connect exclusively to the highest-degree vertices in the graph, rather than to one another, generating a tree-like topology. This particular topology yields a signature in the tail of the spectrum.

2.4.4 The Watts-Strogatz model. Many real-world graphs exhibit the so-called "small world phenomenon," where the expected shortest path length between two vertices chosen uniformly at random grows logarithmically with the size of the graph. Watts and Strogatz [8] constructed a random graph model that exhibits this behavior, along with a high clustering coefficient not seen in an uncorrelated random graph. The clustering coefficient is defined as the ratio of number of triangles to the number of connected triplets of vertices in the graph. The **Watts-Strogatz model** [8] is designed to be the simplest random graph that has high local clustering and small average shortest path distance between vertices.

Like the preferential attachment model, this graph is most easily described via a generative mechanism. The algorithm proceeds as follows. Let n be the size of the desired graph, let $0 \leq p \leq 1$, and let k be an even integer, with $k < n$. We begin with a ring lattice, which is a graph where each vertex is attached to its k nearest neighbors, $k/2$ on each side. We then randomly rewire edges (effectively creating shortcuts) as follows. With probability p, each edge (i, j) (where $i < j$) is replaced by the edge (i, l), where l is chosen uniformly at random. The target l is chosen so that $i \neq l$ and i is not connected to l at the time of rewiring. We stop once all edges have been iterated through. We add an additional stipulation that the graph must be connected. If the algorithm terminates with a disconnected graph, then we restart the algorithm and generate a new graph.

As mentioned before, the topological features that are significant in this graph are the high local clustering and short expected distance between vertices. Of course, these quantities are dependent on the parameter p; as $p \to 1$, the Watts-Strogatz model approaches an uncorrelated random graph. Similarly, as $p \to 1$ the adjacency spectral density transitions from the tangle of sharp maxima typical of a ring-lattice graph to the smooth semi-circle of the uncorrelated random graph [19]. Unlike the models above, this model exhibits primarily local structure. Indeed, we observe that the most significant differences lie in the tail of the adjacency spectrum, that can be directly linked to the number of triangles in the graph [19]. On the large scale, however, this graph looks much like the uncorrelated random graph, in which it exhibits no communities or high-degree vertices.

This model fails to produce the scale-free behavior observed in many real world networks. Although the preferential attachment model reproduces this scale-free behavior, it fails to reproduce the high local clustering that is frequently observed, and so we should think of neither model as fully replicating the properties of observed graphs.

2.4.5 The configuration model. The above three models are designed to mimic certain properties of real world networks. In some cases, however, we may wish to create a random graph with a prescribed degree sequence. That is to say, we seek a distribution that assigns equal probability to each graph, conditioned upon the graph having a given degree sequence. The simplest model that attains this result is the configuration model [84]. Recently, Zhang et al. [85] have derived an asymptotic expression for the adjacency spectrum of a configuration model, that is exact in the limit of large graph size and large mean degree.

Inconveniently, this model is not guaranteed to generate a simple graph; the resulting graph can have self-edges, or multiple edges between two vertices. In 2010, Bayati et al. [86] described an algorithm that samples (approximately) uniformly from the space of simple graphs with a given degree distribution. In [86] the authors prove that the distribution is asymptotically uniform, but they do not prove results for finite graph size (see [87] for a more detailed analysis). We use this algorithm despite the fact that it does not sample the desired distribution in a truly uniform manner; the fact that the resulting graph is simple overcomes this drawback.

We refer to graphs sampled in this way as *configuration model* graphs. The significance of this class of graphs stems from the fact that we can use them to control for the degree sequence when comparing graphs; they are used as a null model, similar to the uncorrelated random graph, but they can be tuned to share some structure (notably, the power-law degree distribution of preferential attachment) with the graphs to which they are compared.

2.4.6 Lattice graphs. We use a 2-dimensional x by y rectangular lattice as a prototypical example of a highly regular graph. This regularity is reflected by the discrete nature of the lattice's spectrum, which can be seen in Fig 1. The planar structure of the lattice allows for an intuitive understanding of the eigenvalues, as they approximate the vibrational frequencies of a two-dimensional surface.

This is a particularly strong flavor of local structure, as it is not subject to the noise present in random graph models. This aspect allows us to probe the functioning of our distances when they are exposed to graphs with a high amount of inherent structure and very low noise.

2.4.7 Exponential random graph models. A popular random graph model is the **exponential random graph model**, or ERGM for short. Although they are popular and enjoy simple interpretability, we do not use ERGMs in our experiments. Unlike some of our other models that are described by their generative mechanisms, these are described directly via the probability of observing a given graph G.

Let $g_i(G)$ be some scalar graph properties (e.g., size, volume, or number of triangles) and let θ_i be corresponding coefficients, for $i = 1, \ldots, K$. Then, the ERGM assigns to each graph a

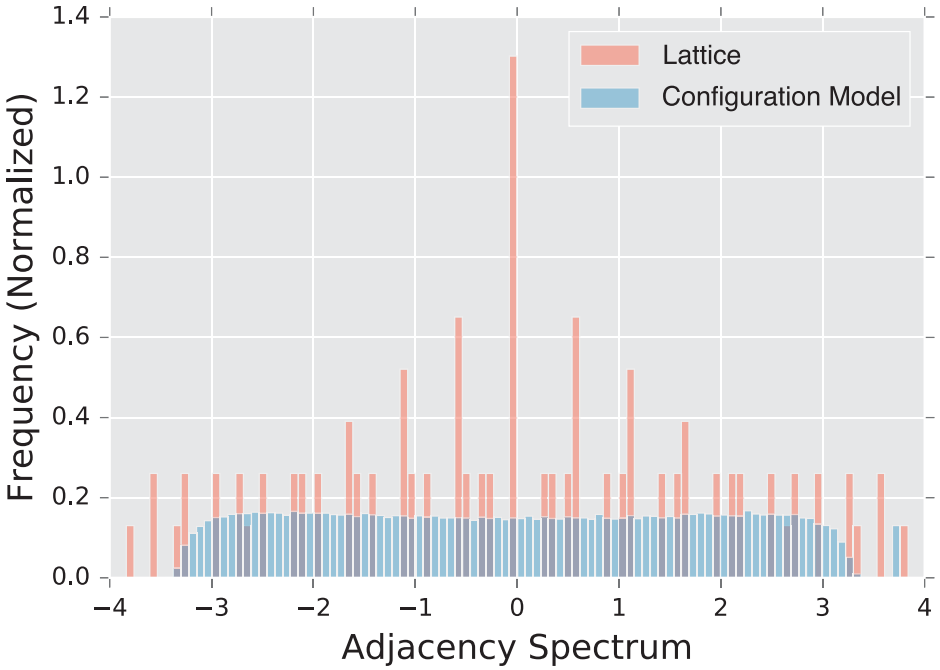

Fig 1. Spectral densities λ^A of the adjacency matrix for a lattice graph and a degree matched configuration model. Densities are built from an ensemble of 1,000 graphs generated using parameters described in Subsection3.1.5.

https://doi.org/10.1371/journal.pone.0228728.g001

probability [88]

$$\mathbb{P}(G) \propto \exp\left(\sum_{i=1}^{K} \theta_i g_i(G)\right).$$

This distribution can be sampled via a Gibbs sampling technique, a process that is outlined in detail in [88]. ERGMs show great promise in terms of flexibility and interpretability; one can seemingly tune the distribution towards or away from any given graph metric, including mean clustering, average path length, or even decay of the degree distribution.

However, our experience attempting to utilize ERGMs led us away from this approach. When sampling from ERGMs, we were unable to control properties individually to our satisfaction. We found that attempts to increase the number of triangles in a graph increased the graph volume; when we subsequently used the ERGM parameters to *de-emphasize* graph volume, the sampled graphs had an empirical distribution very similar to an uncorrelated random graph.

2.4.8 Graph neural networks. Each one of the graph ensembles described in the previous sections represents the quintessential exemplar of a certain graph *structure* (e.g, degree distribution, clustering coefficients, shortest path distance, community structure, etc.) Each ensemble can be thought as the atomic building block that can be used to understand complex existing real world networks. For instance, it is shown in [89] that any sufficiently large graph behaves approximately like a stochastic blockmodel. These networks are also amenable to a rigorous mathematical analysis, and one can analyze the influence on the graph distances of changes in the graph geometry and topology.

As explained in section 2.2.4, there has been some very recent attempts at generating random realizations of graphs by learning the structure of massive datasets of existing networks

(e.g, [90–92]). These algorithms offer an implicit representation of a set of graphs, by discovering an optimal neural network that can generate new graphs with similar structures. In contrast to the prototypical random graph ensembles, the current understanding of the theoretical properties of the graph neural networks is very limited: there are no results on the structural properties of these models (but see [93] for an estimate of the complexity of a graph convolutional network (number of nodes and number of hidden units) required to learn graph moments).

A systematic study of the sensitivity of graph distances on graph neural networks is clearly needed. Such a study would provide information that would complement theoretical results that elucidate how expressive such graph models can be [55, 56]. Unfortunately, such experiments clearly go beyond the scope of the current manuscript.

2.5 Real world networks

Random graph models are often designed to simulate a single important feature of real world networks, such as clustering in the Watts-Strogatz model or the high-degree vertices of the preferential attachment model. In real networks, these factors coexist in an often unpredictable configuration, along with significant amounts of noise. Although the above analysis of the efficacy of various distances on random graph scenarios can help inform and guide our intuition, to truly understand their utility we must also look at how they perform when applied to empirical graph data.

In this study, we evaluate the performance of the aforementioned distances using two scenarios. First, we study the change point detection scenario for two time-varying networks: a dynamic social-contact graph, collected via RFID tags in an French primary school [94], and a time series of emails exchanged between 986 members of a large European research institution [95] over a period of 803 days.

Secondly, we investigate the two-sample test problem in neuroscience: given two populations of functional brain connectivity networks, we compute a statistic to test whether both populations are generated by the same probability distribution of controls (null hypothesis), or one population is significantly different from the other one. Specifically, we compare the functional connectivity of subjects with a diagnosis of autism spectrum disorder [96] versus a population of controls.

2.5.1 Primary school face to face contact. Some of the most well-known empirical network datasets reflect social connective structure between individuals, often in online social network platforms such as Facebook and Twitter. These networks exhibit structural features such as communities and highly connected vertices, and can undergo significant structural changes as they evolve in time. Examples of such structural changes include the merging of communities, or the emergence of a single user as a connective hub between disparate regions of the graph.

Description of the Experiment. The data are part of a study of face to face contact between primary school students [94]. Briefly, RFID tags were used to record face-to-face contact between students in a primary school in Lyon, France in October, 2009. Events punctuate the school day of the children (see Table 3), and lead to fundamental topological changes in the contact network (see Fig 2). The school is composed of ten classes: each of the five grades (1 to 5) is divided into two classes (see Fig 2).

The construction of a dynamic graph proceeds as follows: time series of edges that correspond to face to face contact describe the dynamics of the pairwise interactions between students. We divide the school day into $N = 150$ time intervals of $\Delta t \approx 200$ s. We denote by $t_i = 0, \Delta t, \ldots, (N-1)\Delta t$, the corresponding temporal grid. For each t_i we construct an undirected

Table 3. Events that punctuate the school day.

Time	Event
10:30 a.m.–11:00 a.m.	Morning Recess
12:00 p.m.–1:00 p.m.	First Lunch Period
1:00 p.m.–2:00 p.m.	Second Lunch Period
3:30 p.m.–4:00 p.m.	Afternoon Recess

https://doi.org/10.1371/journal.pone.0228728.t003

unweighted graph G_{t_i}, where the $n = 232$ nodes correspond to the 232 students in the 10 classes, and an edge is present between two students u and v if they were in contact (according to the RFID tags) during the time interval $[t_{i-1}, t_i)$.

For the purpose of this work, we think of each class as a community of connected students; classes are weakly connected (e.g., see Fig 2 at times 9:00 a.m., and 2:03 p.m.). During the school day, events such as lunchtime and recess, trigger significant increases in the the number of links between the communities, and disrupt the community structure; see Fig 2 at times 11:57 a.m., and 1:46 p.m.

2.5.2 European Union Emails. **Description of the Data**. The data were obtained from the Stanford Large Network Dataset Collection [97]. The network was generated using anonymized emails exchanged between 986 members of a large European research institution [95]. There are 986 nodes that correspond to distinct individuals sending and receiving emails. To

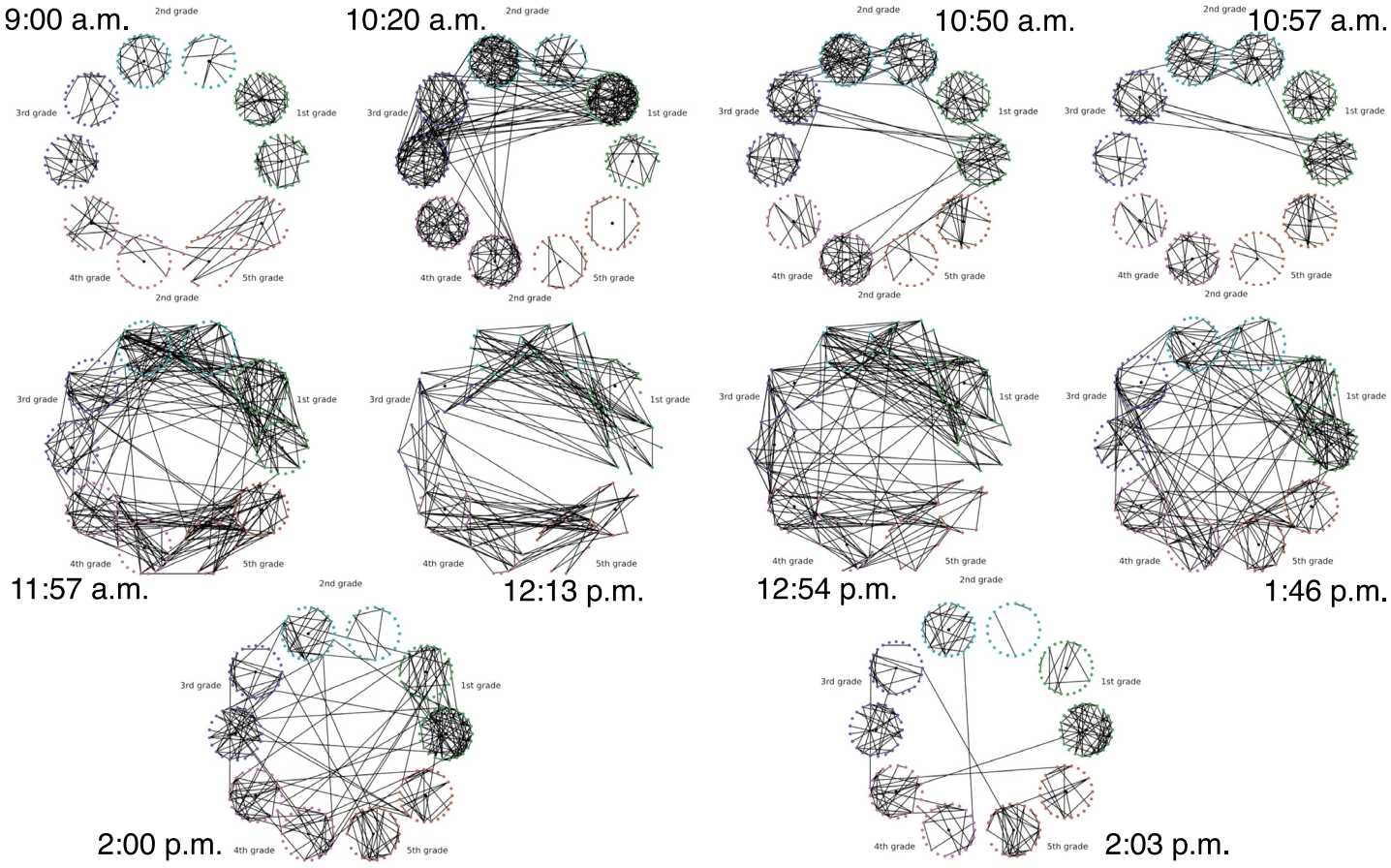

Fig 2. Top to bottom, left to right: Snapshots of the face-to-face contact network at times (shown next to each graph) surrounding significant topological changes.

https://doi.org/10.1371/journal.pone.0228728.g002

reduce the variability in the data, we aggregate the emails exchanged every week, and perform an analysis at the week level. An edge was created between nodes i and j if both i sent at least one email to j and j sent at least one email to i during that week. The timeline starts on October 1, 2003 [95]. The graph distances were computed between the weekly graphs thus created.

2.5.3 Functional brain connectivity. Graph theoretical analysis of the connective structure of the human brain is a popular research topic, and has benefited from our growing ability to analyze network topology [98–100]. In these graph representations of the brain, the vertices are physical regions of the brain, and the edges indicate the connectivity between two regions. The connective structure of the brain is examined either at the "structural" level, in which edges represent anatomical connection between two regions, or at the "functional" level, in which an edge connects regions whose activation patterns are in some sense similar. Developmental and mental disorders such as autism spectrum disorder [101] and schizophrenia [102] have been shown to have structural correlates in the graph representations of the brains of those affected. In this study we focus on autism spectrum disorder, or ASD.

Description of the Data. The Autism Brain Imagine Data Exchange [96, 103], or ABIDE, is an aggregation of brain-imaging data sets from laboratories around the world that study ASD [96]. The data that we focus on are measurements of the activity level in various regions of the brain, measured via functional magnetic resonance imaging (fMRI).

After preprocessing, the data is analyzed for quality. Of the original 1114 subjects (521 ASD and 593 TD), only 871 pass this quality-assurance step. These subjects are then spatially aggregated via the Automated Anatomical Labelling (AAL) atlas, that aggregates the spatial data into 116 time series.

To construct a graph from these time series, the pairwise Pearson correlation is calculated to measure similarity. If we let u and v denote two regions in the AAL atlas and let $\rho(u, v)$ denote the Pearson correlation between the corresponding time series, the simplest way to build a graph is to assign weights $w(u, v) = |\rho(u, v)|$. We exclude low correlations, as these are often spurious and not informative as to the structure of the underlying network, and define the weights

$$w(u,v) = \begin{cases} |\rho(u,v)| & |\rho(u,v)| \geq T \\ 0 & |\rho(u,v)| < T. \end{cases}$$

Finally, we also construct an unweighted graph according to

$$w(u,v) = \begin{cases} 1 & |\rho(u,v)| \geq T \\ 0 & |\rho(u,v)| < T. \end{cases}$$

We will compare both binary and weighted connectomes, generated for multiple thresholds. This will allow us to be confident that our results are not artifacts of poorly chosen parameters in our definition of the connectome graph.

2.6 Evaluation protocol: The distance contrast

2.6.1 The distance contrast. The experiments are designed to mimic a scenario in which a practitioner is attempting to determine whether a given graph belongs to a population or is an outlier relative to that population.

Specifically, let us define by $\mathcal{G}_0$ and $\mathcal{G}_1$ two graph populations, which we refer to as the **null** and **alternative** populations respectively. For each distance measure, let $\mathcal{D}_0$ be the distribution of distances $d(G_0, G_0')$ where G_0 and G_0' are both drawn from the distribution $\mathcal{G}_0$. Similarly, let

$\mathcal{D}_1$ be the distribution of distances $d(G_0, G_1)$, where G_0 is drawn from $\mathcal{G}_0$ and G_1 is drawn from $\mathcal{G}_1$.

The statistic $\mathcal{D}_0$ characterizes the natural variability of the graph population $\mathcal{G}_0$, as seen through the lens formed by the distance d. Similarly, the statistic $\mathcal{D}_1$ reveals how distant—according to the distance d—the two graph populations $\mathcal{G}_0$ and $\mathcal{G}_1$ are. If the distributions of $\mathcal{D}_0$ and $\mathcal{D}_1$ are well separated, then d is effective at differentiating the null population from the alternative population.

To that end, we normalize the statistics of $\mathcal{D}_1$ by those of $\mathcal{D}_0$ in order to compare. In particular, let μ_i be the sample mean of $\mathcal{D}_i$, and let σ_i be the sample standard deviation, for $i \in \{0, 1\}$. We define the following (normalized) contrast, $\hat{\mathcal{D}}_1$, [104], between D_0 and D_1, whose samples $\hat{D}_1$ are calculated via

$$\hat{D}_1 \stackrel{\text{def}}{=} \frac{D_1 - \mu_0}{\sigma_0}. \tag{12}$$

This studentized distance contrast can also be related to the Wald test statistic [104]

$$\frac{D_0 - \mu_0}{\sigma_0}. \tag{13}$$

If the empirical distribution of contrast $\hat{\mathcal{D}}_1$ is well separated from zero, viz. the contrast between D_1 and the sample mean μ_0 is significantly greater than the standard deviation, then the distance is effectively separating the null and alternative populations.

2.6.2 Comparisons of the random graph ensembles. Table 4 describes the various experiments. Each model is compared against a null model; the null model can be sampled either from the Erdős-Rényi model, or from a configuration model. The latter makes it possible to match the degree distribution of the model being tested against that of the null model. We compare the distance contrast in (12) between each model and the null model using all the distances. When appropriate, we also report the performance of the spectral distances for various k. Table 4 also displays the structural feature that is being evaluated for a particular experiment.

2.6.3 Comparisons of the real world networks. Primary School Face to Face Contact and European Union Emails. Temporal changes in the graph topology over time are quantified using the various distance measures. For each distance measure d, we defined the following temporal difference,

$$D_R(t_i) \stackrel{\text{def}}{=} d(G_{t_{i-1}}, G_{t_i}).$$

To help compare these distances with one another, we normalize each by its sample mean

Table 4. Table of comparisons performed, and the important structural features therein. $G(n, p)$ indicates the Erdős-Rényi uncorrelated random graph, SBM is the stochastic blockmodel, PA is the preferential attachment model, CM is the degree matched configuration model, and WS is the Watts-Strogatz model.

Section	Null	Alternative	Structural Difference
Stochastic Block Model	$G(n, p)$	SBM	Community structure
Preferential Attachment	$G(n, p)$	PA	High-degree vertices
Preferential Attachment vs Configuration Model	CM	PA	Structure not in the degree sequence
Watts-Strogatz	$G(n, p)$	WS	Local structure
Lattice Graph	$G(n, p)$	Lattice	Extreme local structure

https://doi.org/10.1371/journal.pone.0228728.t004

$\bar{D} = N^{-1}\sum_i D(t_i)$, and we define the normalized temporal difference,

$$\hat{D}(t) = D(t)/\bar{D}.$$

Functional Brain Connectivity. We define $\mathcal{G}_1$ to be the set of connectomes computed from the ASD subjects, and $\mathcal{G}_0$ the set of connectomes from the control population (null model). The evaluation proceeds as described in Section 2.6. The distance contrast between the two populations is evaluated using the statistic defined in (12),

$$\hat{D}_1 \stackrel{\text{def}}{=} \frac{D_1 - \mu_0}{\sigma_0}. \tag{14}$$

3 Results

3.1 Random graph ensembles

For each experiment described in Table 4, we generate 50 samples of D_0 and D_1, where each sample compares two graphs of size $n = 1,000$, unless otherwise specified.

The graphs are always connected; the sampler will discard a draw from a random graph distribution if the resulting graph is disconnected. Said another way, we draw from the distribution defined by the model, conditioning upon the fact that the graph is connected.

The small size of the graphs allows us to use larger sample sizes; although all of the matrix distances used have fast approximate algorithms available, we use the slower, often $\mathcal{O}(n^2)$, exact algorithms for the experiments, and so larger graphs would be prohibitively slow to work with. In all the experiments, we choose parameter values so that the expected volume of the two models under comparison is equal.

We display the performance of the various distances on the same figure. Boxes extend from lower to upper quartile, with center line at median. Whiskers extend from 5th to 95th percentile (e.g., see Fig 3). We also display the performance of the spectral distances contrast $\hat{D}_1$—for the three matrices: adjacency, combinatorial Laplacian, and normalized Laplacian as a function of the number of eigenvalues used to compute the distance (e.g., see Fig 4).

3.1.1 Stochastic blockmodel. Fig 3 displays the comparison between a stochastic blockmodel and an uncorrelated random graph model (null model). The edge density $p = 0.12$ of the uncorrelated random graph is chosen so that graphs are connected with high probability. With this choice of parameters, we observe that the empirical probability of generating a disconnected uncorrelated random graph with these parameters is $\sim 0.02\%$. The preferential attachment section describes in more detail why this exact value is chosen.

The stochastic blockmodel is composed of two communities of equal size, $n/2 = 500$. Stochastic blockmodel experiments are run with in-community parameter $p = 1.9 \times 0.02$, and cross-community parameter $q = 0.1 \times 0.02$. Thus, the in-community connectivity is denser than the cross-community connectivity by a factor of $p/q = 19$.

Since we have matched the volume of the graphs, the edit distance fails to distinguish the two models. Among the matrix distances, DeltaCon separates the two models most reliably. The adjacency and normalized Laplacian distances perform well. The resistance perturbation distance and the non-normalized Laplacian distance fail to distinguish the two models.

As confirmed in Fig 4, the performance of the adjacency distance is primarily driven by differences in the second eigenvalue λ_2^A, and including further eigenvalues adds no benefit; the normalized Laplacian also shows most of its benefit in the second eigenvalue $\lambda_2^{\mathcal{L}}$, but unlike the adjacency distance, including more eigenvalues decreases the performance of the metric.

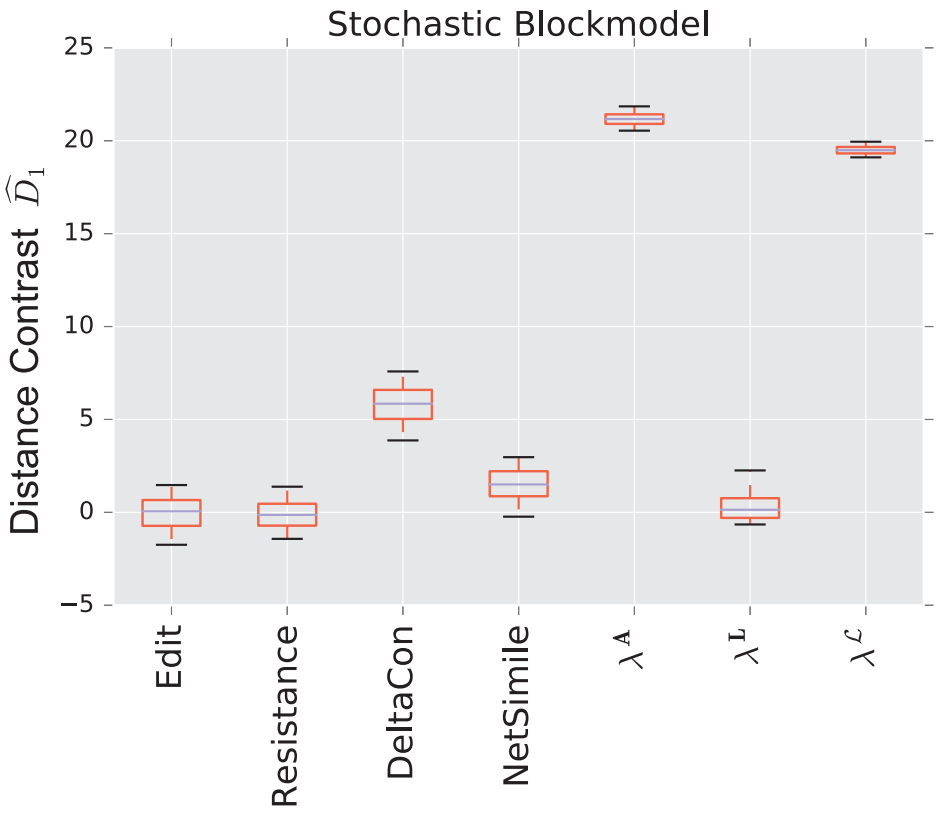

Fig 3. Distance contrast $\hat{D}_1$ between the stochastic blockmodel and the uncorrelated random graph model (null model).

https://doi.org/10.1371/journal.pone.0228728.g003

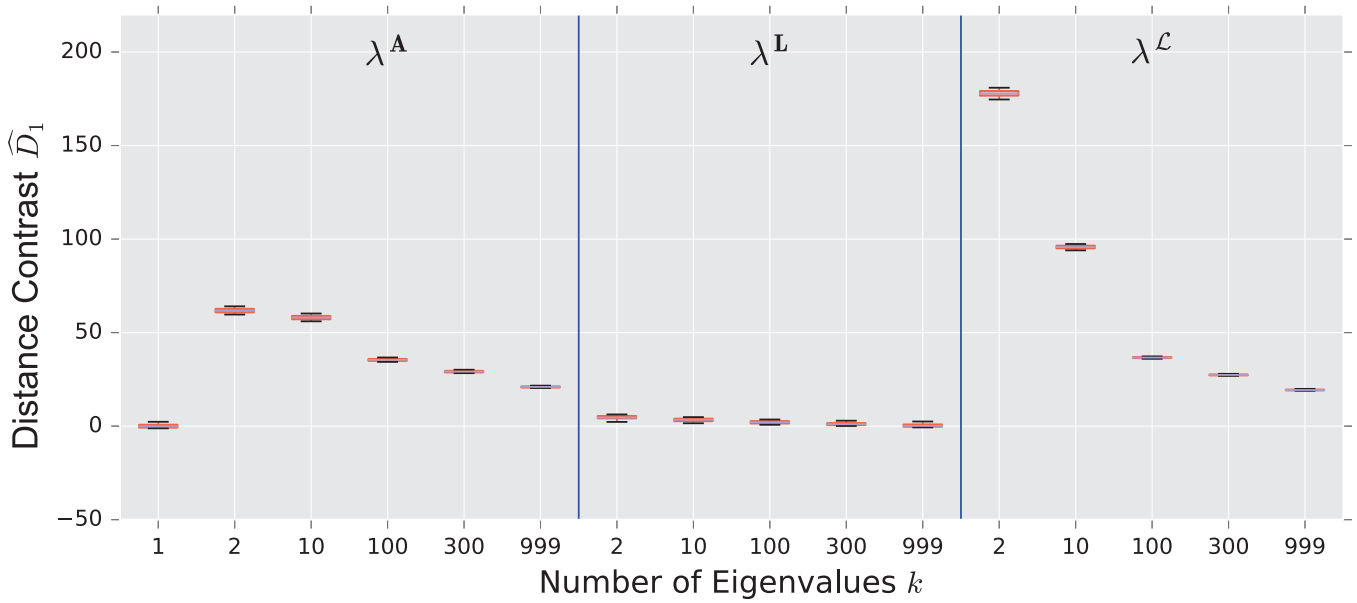

Fig 4. Spectral distances contrast $\hat{D}_1$—For the three matrices: Adjacency, combinatorial Laplacian, and normalized Laplacian (from left to right)—Between the stochastic blockmodel and the uncorrelated random graph model (null model).

https://doi.org/10.1371/journal.pone.0228728.g004

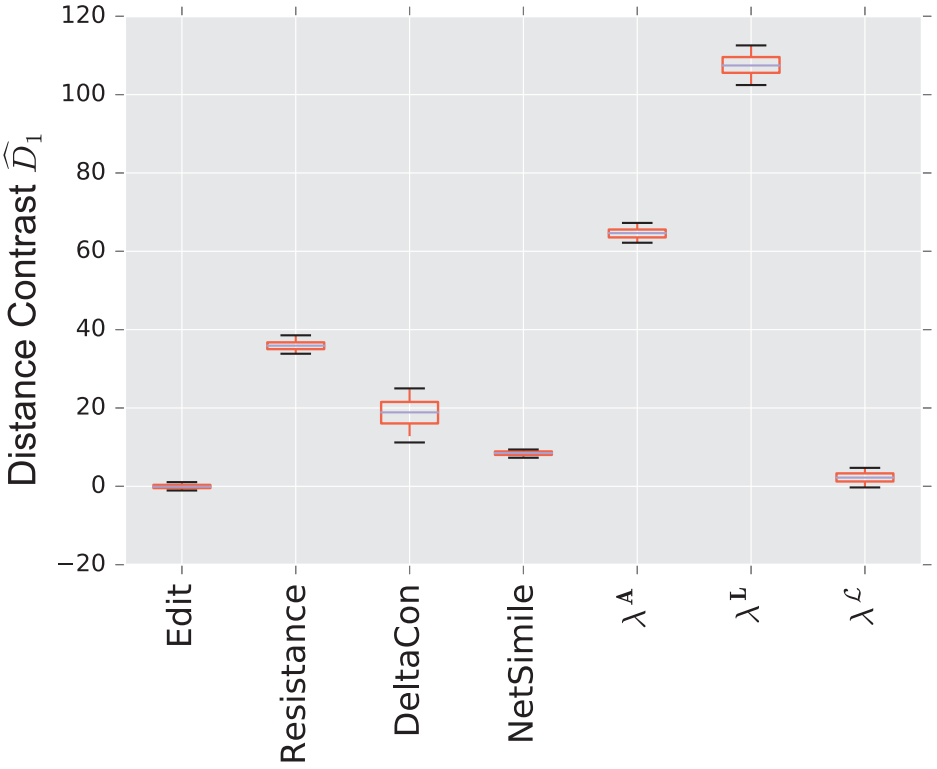

Fig 5. Distance contrast $\hat{D}_1$ between the preferential attachment and the uncorrelated random graph model (null model).

https://doi.org/10.1371/journal.pone.0228728.g005

3.1.2 Preferential attachment vs uncorrelated. Fig 5 shows the results of comparing a preferential attachment graph to an uncorrelated random graph. The preferential attachment graph is quite dense, with $l = 6$. Since the number of edges in this model is always $|E| = l(n - l)$, we determine the parameter p for the uncorrelated graph via

$$p(l) = l(n-l)\binom{n}{2}^{-1},$$

to guarantee that both graphs always have the same volume.

Again, because both graphs have the same number of edges (with high probability), the edit distance fails to distinguish the two models. The resistance distance shows mediocre performance, although 0 is outside the 95% confidence interval. DeltaCon exhibits extremely high variability, although it has the highest median of the matrix distances.

The combinatorial Laplacian distance outperforms all others, while the normalized Laplacian does not separate the two models at all. Fig 6 shows that the very fine scale eigenvalues of the combinatorial Laplacian (large index) are needed to differentiate the two models. Conversely, the discriminating eigenvalues of the adjacency matrix are the smallest eigenvalue; in fact, the first eigenvalue captures much of the contrast: the distance contrast (12) stays more or less constant as one increases k (see Fig 6).

3.1.3 Preferential attachment vs configuration model. To further explore the distinctive features of the preferential attachment graphs, we change here the null model. Instead of using a volume matched uncorrelated random graph model, we use a degree matched configuration

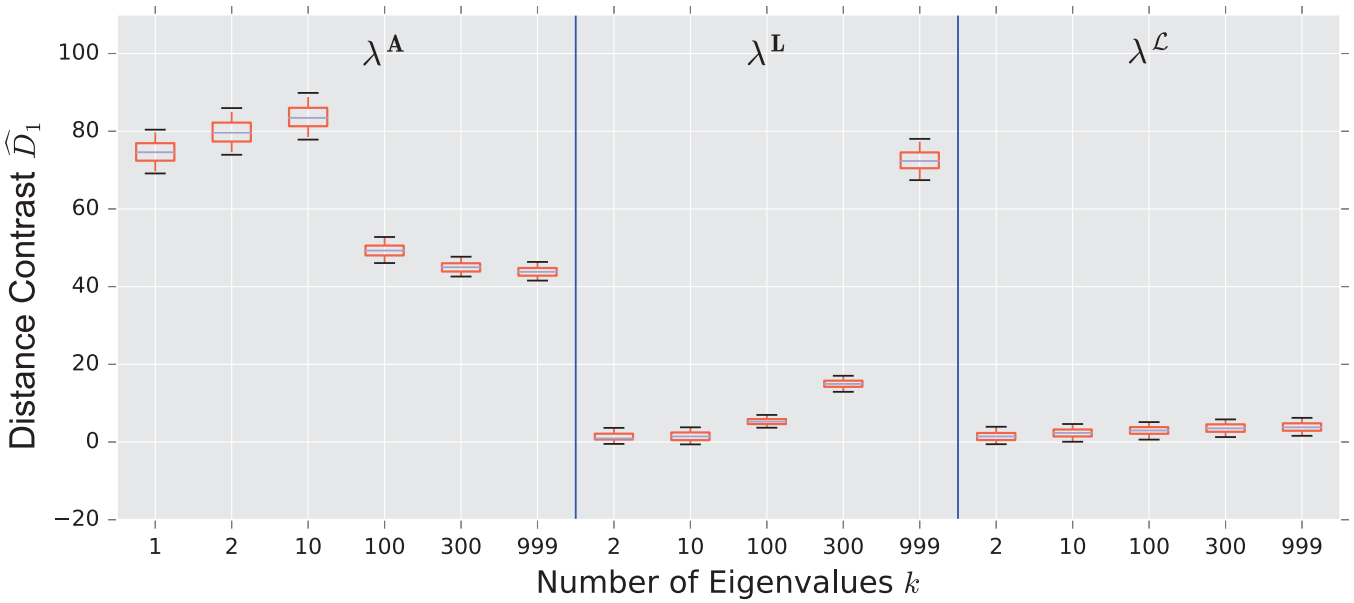

Fig 6. Spectral distance contrast $\hat{D}_1$—For the three matrices: Adjacency, combinatorial Laplacian, and normalized Laplacian (from left to right)—Between the preferential attachment and the uncorrelated random graph model (null model).

https://doi.org/10.1371/journal.pone.0228728.g006

model as the null model (the volume is automatically matched, since the number of edges is half of the sum of the degrees). This experiment allows us to search for structure in the preferential attachment model that is *not* prescribed by the degree distribution.

An intriguing result happens: not a single distance can differentiate between a preferential attachment graph and a randomized graph with the same degree distribution (see Fig 7).

The spectral distance based on the eigenvalues combinatorial Laplacian $\lambda^{\boldsymbol{L}}$, which yields the strongest contrast when comparing the preferential attachment model to the uncorrelated random graph model is now unavailing. This thought-provoking experiment suggests that all significant structural features of the preferential attachment model are prescribed by the degree distribution.

3.1.4 Watts-Strogatz. The Watts-Strogatz experiments used $k = 20$, $p = 0.020020\ldots$ (calculated so that the volumes match) and $\beta = 0.1$. The number of nodes was $n = 1,000$.

The Watts-Strogatz model is sparse, and thus the uncorrelated random graph has a low value of p—since we match the number of edges—and is very likely disconnected. This is only a significant problem for the resistance distance, that is undefined for disconnected graphs. To remedy this, we use an extension of the resistance distance called the **renormalized resistance distance**, that is developed and analyzed in [29]. This is the only experiment in which the use of this particular variant of the resistance distance is required.

Fig 8 shows that the spectral distances based on the adjacency and normalized Laplacian are the strongest performers. Amongst the matrix distances, DeltaCon strongly outperforms the resistance distance. The resistance distance here shows a negative median, that indicates smaller distances between populations than within the null population. This is likely due to the existence of many (randomly partitioned) disconnected components within this particular null model, that inflates the distances generated by the renormalized resistance distance. It is notable that, contrary to the comparison in Section 3.1.2, the normalized Laplacian outperforms the combinatorial Laplacian.

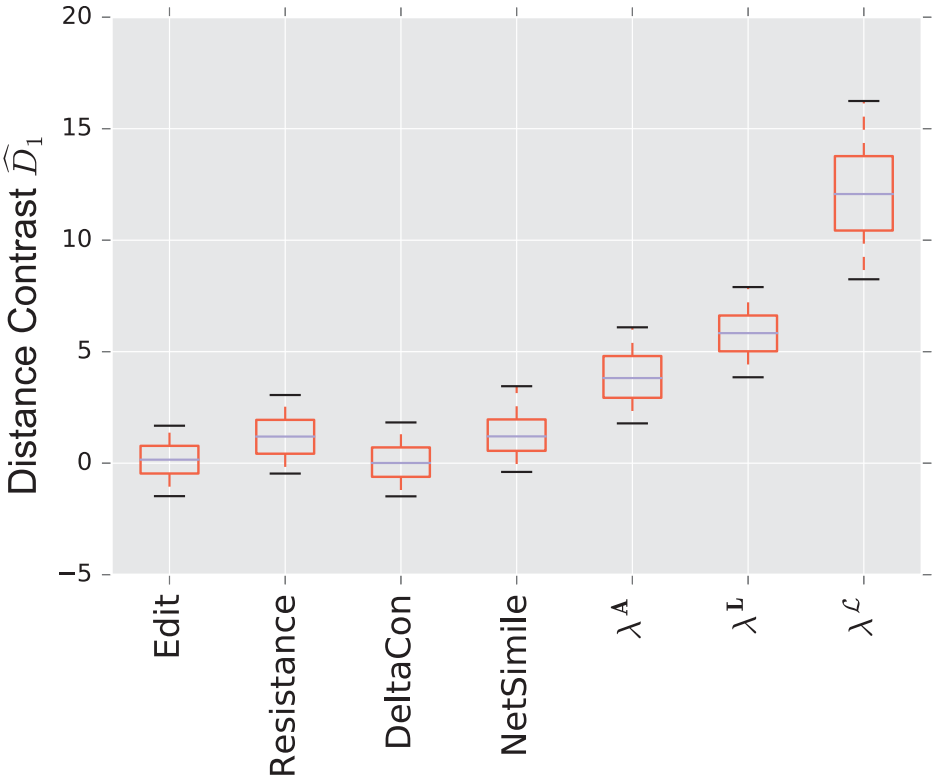

Fig 7. Distance contrast $\hat{D}_1$ between the preferential attachment model and the degree matched configuration model (null model).

https://doi.org/10.1371/journal.pone.0228728.g007

Fig 9 displays the results for spectral distances, for a wide variety of k. This figure is significant because it illustrates the fact that both coarse scales (large k for $\lambda_k^{\mathbf{A}}$) and fine scales (large k for $\lambda_k^{\mathcal{L}}$) are necessary to yield the optimal contrast between the two models.

3.1.5 Lattice graph. The final experiment, compares a lattice graph to a configuration model graph with the same degree distribution.

The lattice graphs are 100×10, giving a total size of 1, 000.

The lattice here is highly structured, while the configuration model graph is quite similar to an uncorrelated random graph; both the deterministic degree distribution of the lattice and the binomial distribution of the uncorrelated random graph are highly concentrated around their respective means.

We see in Fig 10 that the scaled distances in this experiment are about an order of magnitude higher than they are in other experiments for some of the distances; because the lattice is such an extreme example of regularity, it is quite easy for many of the distances to discern between these two models. The resistance distance has the highest performance, while spectral distances all perform equally well. Note that for a regular graph, the eigenvalues of $\boldsymbol{A}$, $\boldsymbol{L}$, and $\mathcal{L}$ are all equivalent, up to an overall scaling and shift, so we would expect near-identical performance for graphs that are nearly regular.

The spectral distances need all the scales (i.e. all the eigenvalues) to discern between the lattice and the configuration models (see Fig 11). This phenomenon, which is similar to the Watts-Strogatz model (see Section 3.1.4), points to the importance of the local structure in the topology of the lattice graph.

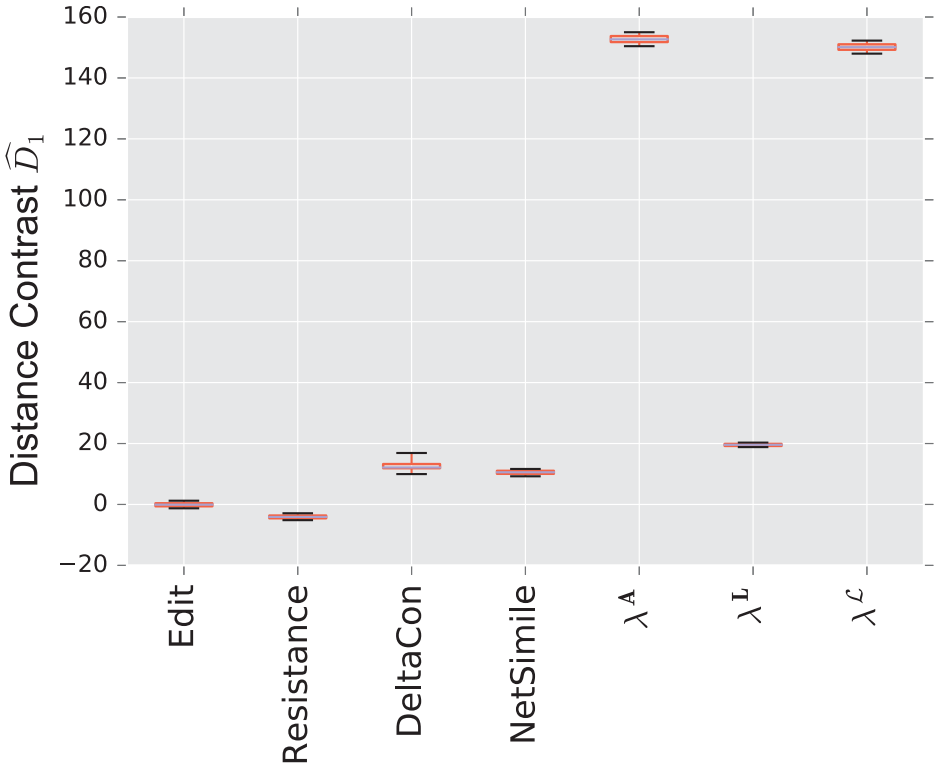

Fig 8. Distance contrast $\hat{D}_1$ between a small-world graph and the the degree matched configuration model (null model).

https://doi.org/10.1371/journal.pone.0228728.g008

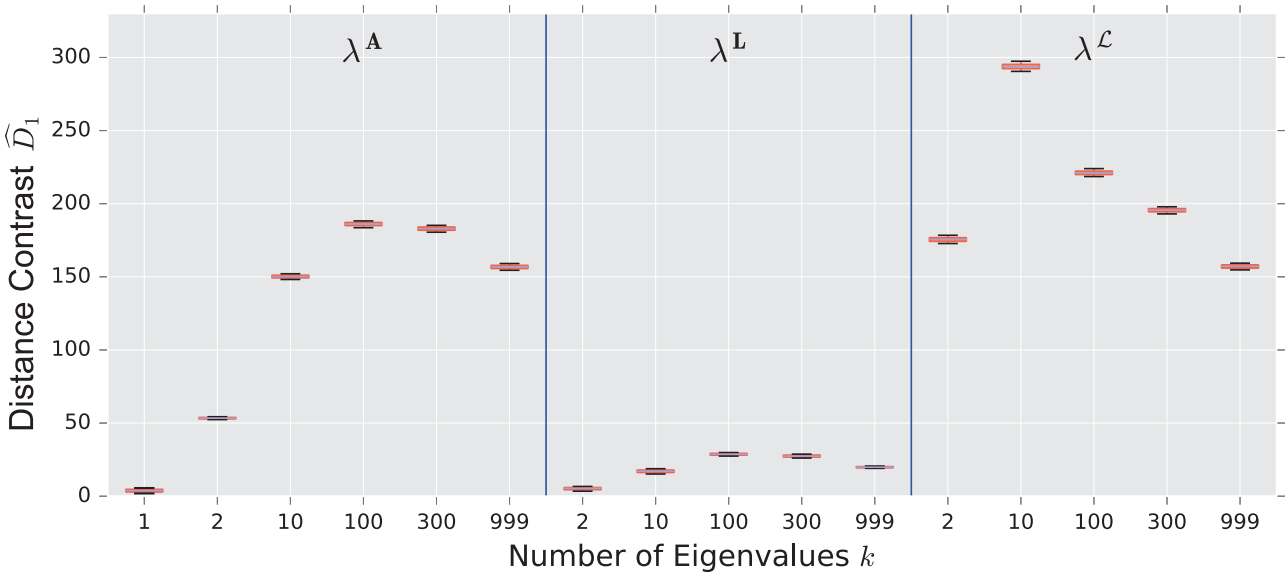

Fig 9. Spectral diastances contrast $\hat{D}_1$—For the three matrices: Adjacency, combinatorial Laplacian, and normalized Laplacian (from left to right)—Between the small-world graph and the degree matched configuration model (null model).

https://doi.org/10.1371/journal.pone.0228728.g009

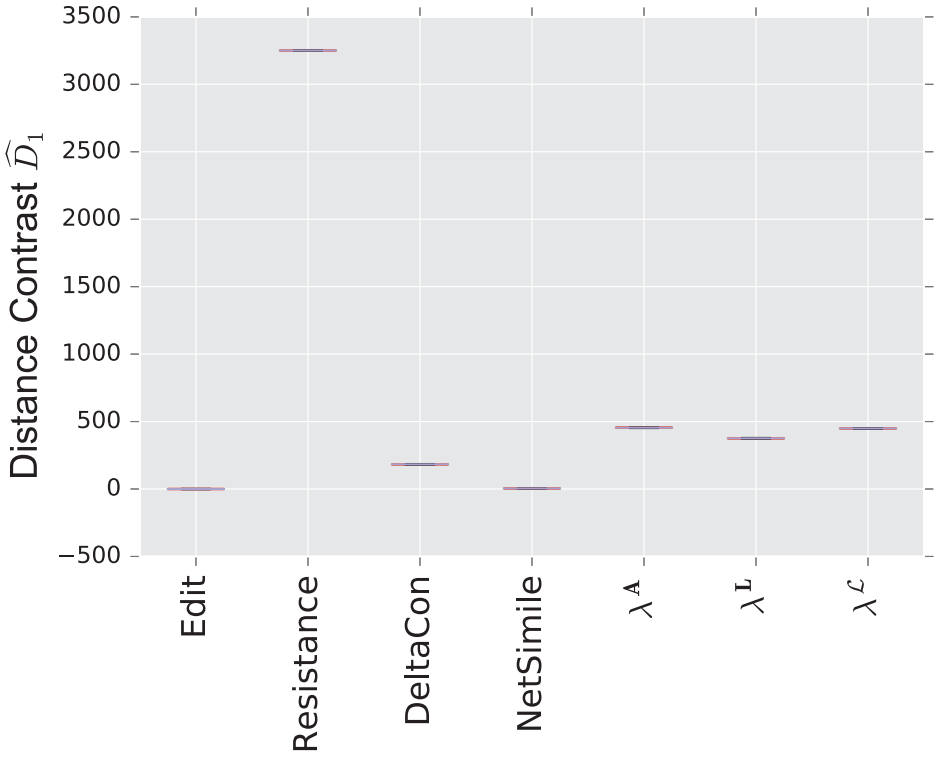

Fig 10. Distance contrast $\hat{D}_1$ between the 10 × 100 two-dimensional lattice graph and the the degree matched configuration model (null model).

https://doi.org/10.1371/journal.pone.0228728.g010

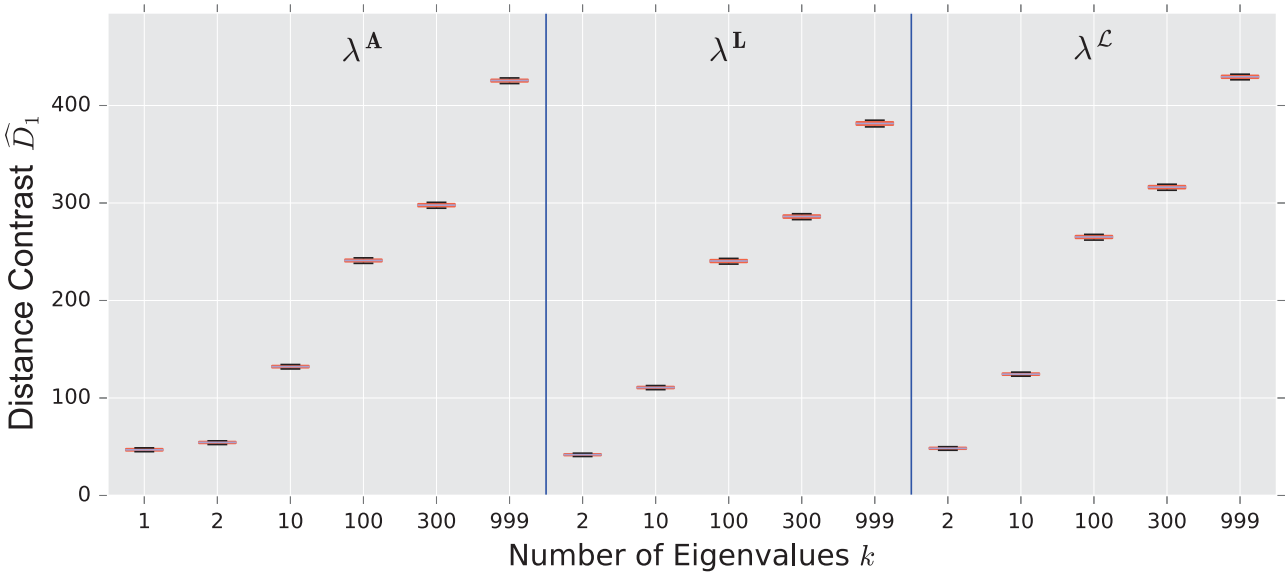

Fig 11. Spectral distances contrast $\hat{D}_1$—For the three matrices: Adjacency, combinatorial Laplacian, and normalized Laplacian (from left to right)—Between the 10 × 10 two-dimensional lattice graph and the degree matched configuration model (null model).

https://doi.org/10.1371/journal.pone.0228728.g011

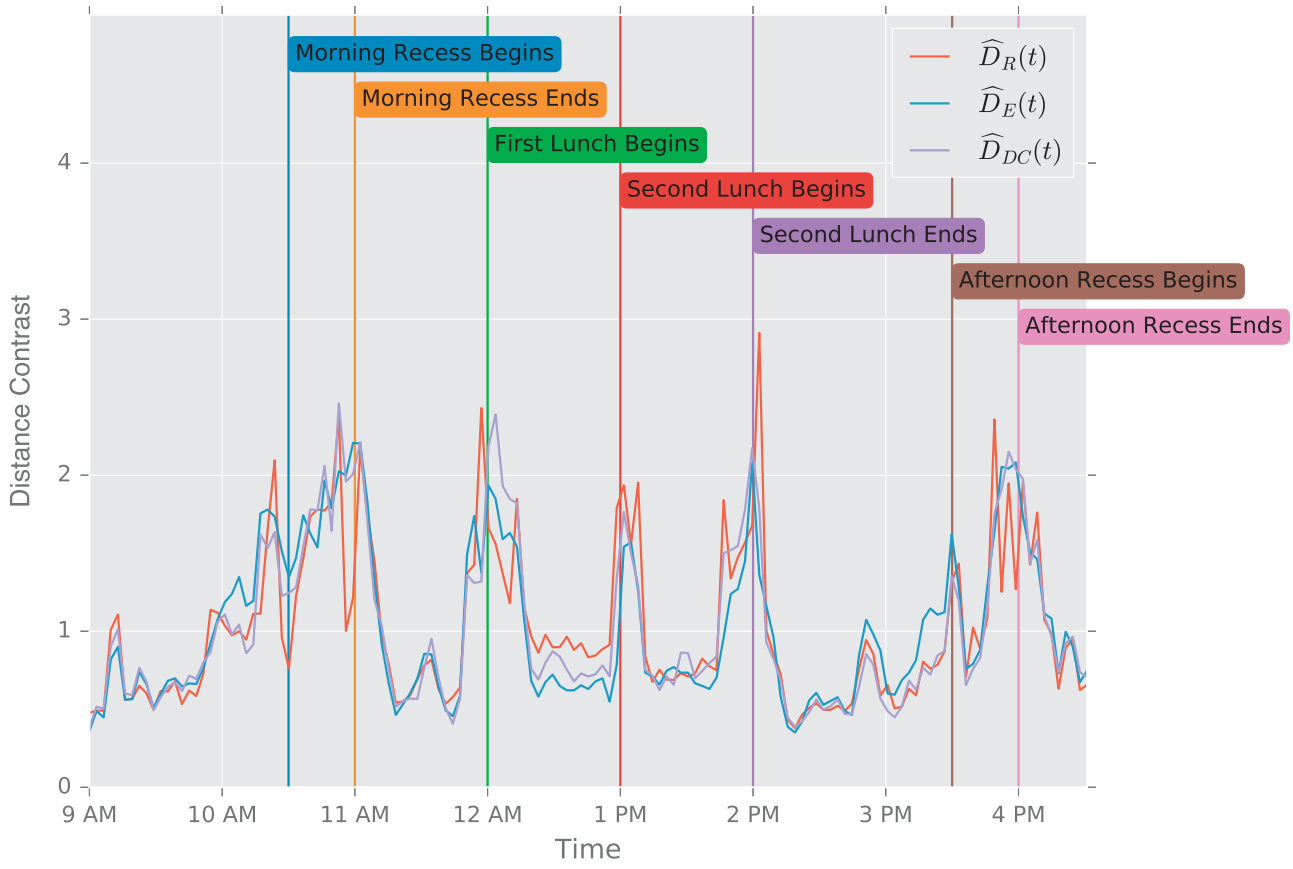

Fig 12. Primary school data set: Normalized temporal differences for the resistance distance $\hat{D}_R$, edit distance $\hat{D}_E$, and DeltaCon distance $\hat{D}_{DC}$.

https://doi.org/10.1371/journal.pone.0228728.g012

3.2 Real world networks

3.2.1 Primary school face to face contact. Fig 12 displays the normalized temporal differences for the resistance distance $\hat{D}_R$, edit distance $\hat{D}_E$, and DeltaCon distance $\hat{D}_{DC}$. All the matrix distances are capable of detecting significant changes in the hidden events that control the topology of the contact network during the school day (see Fig 2). Indeed, the main structural changes that the graph undergoes are transitions into and out of a strong ten-community structure that reflects the classrooms of the school. For example, the adjacency matrix begins as (mostly) block-diagonal at 9 AM, but has significant off-diagonal elements by morning recess at 10:20 AM, and is no longer (block) diagonally dominant come the lunch period at 12 PM.

There exists a persistent random variability of the very fine scale connectivity (e.g., edges come and go within a community) that is superimposed on the large scale structural changes. Unlike, the matrix distances (displayed in Fig 12), NetSimile is significantly affected by these random fluctuations (see Fig 13).

The stochastic variability in the connectivity appreciably influence the high frequency (fine scale) eigenvalues. Consequently, the spectral distances, which are computed using all the eigenvalues, lead to very noisy normalized temporal differences (see Fig 14), making it difficult to detect the significant changes in the graph topology triggered by the school schedule.

3.2.2 European Union Emails. Fig 15 displays the changes in the volume of subsequent graphs (difference in the total number of symmetric emails between two weeks), along with

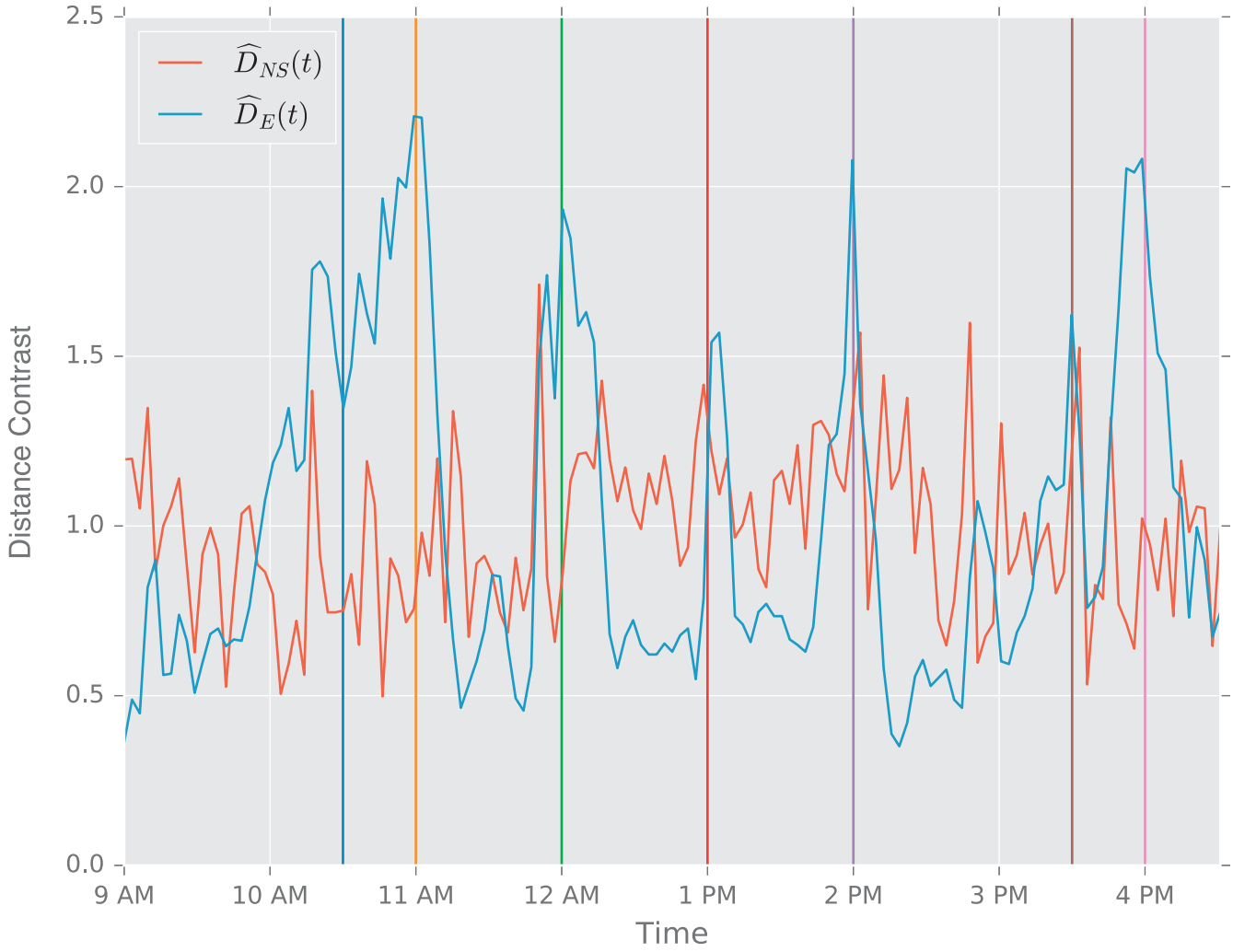

Fig 13. Primary school data set: Normalized temporal differences for the NetSimile distance $\hat{D}_{NS}$ and edit distance $\hat{D}_{E}$.

https://doi.org/10.1371/journal.pone.0228728.g013

the edit distance, as a function of time. Inspired by the analysis of the dataset performed by [105], we superimposed some events that are related to the activity of the European Parliament. These events were retrieved from [106], and are displayed in Table 5.

Fig 16 displays the normalized temporal differences for the NetSimile distance $\hat{D}_{NS}$ and the edit distance $\hat{D}_{E}$. Fig 17 displays the normalized temporal differences for three spectral distances: combinatorial Laplacian $\hat{D}_{L}$, normalized Laplacian $\hat{D}_{\mathcal{L}}$, and adjacency $\hat{D}_{A}$. All the spectral distances are correlated to the activity of the Parliament, including the hearings sessions and the entry into office of the new 2004-2009 Commission, at the end of November 2004. Fig 18 displays the normalized temporal differences for the resistance distance $\hat{D}_{R}$, edit distance $\hat{D}_{E}$, and DeltaCon distance $\hat{D}_{DC}$. All the matrix distances are capable of detecting the election of Jose Barroso as President of the Commission, as well as the investiture procedure of the 2004-2009 Commission: hearings in October 2004, and entry into office at the end of November 2004.

3.2.3 Functional brain connectivity. As explained in Section 2.5.3, the comparisons between sets of connectomes lead to two types of analysis: the analysis of weighted (by the

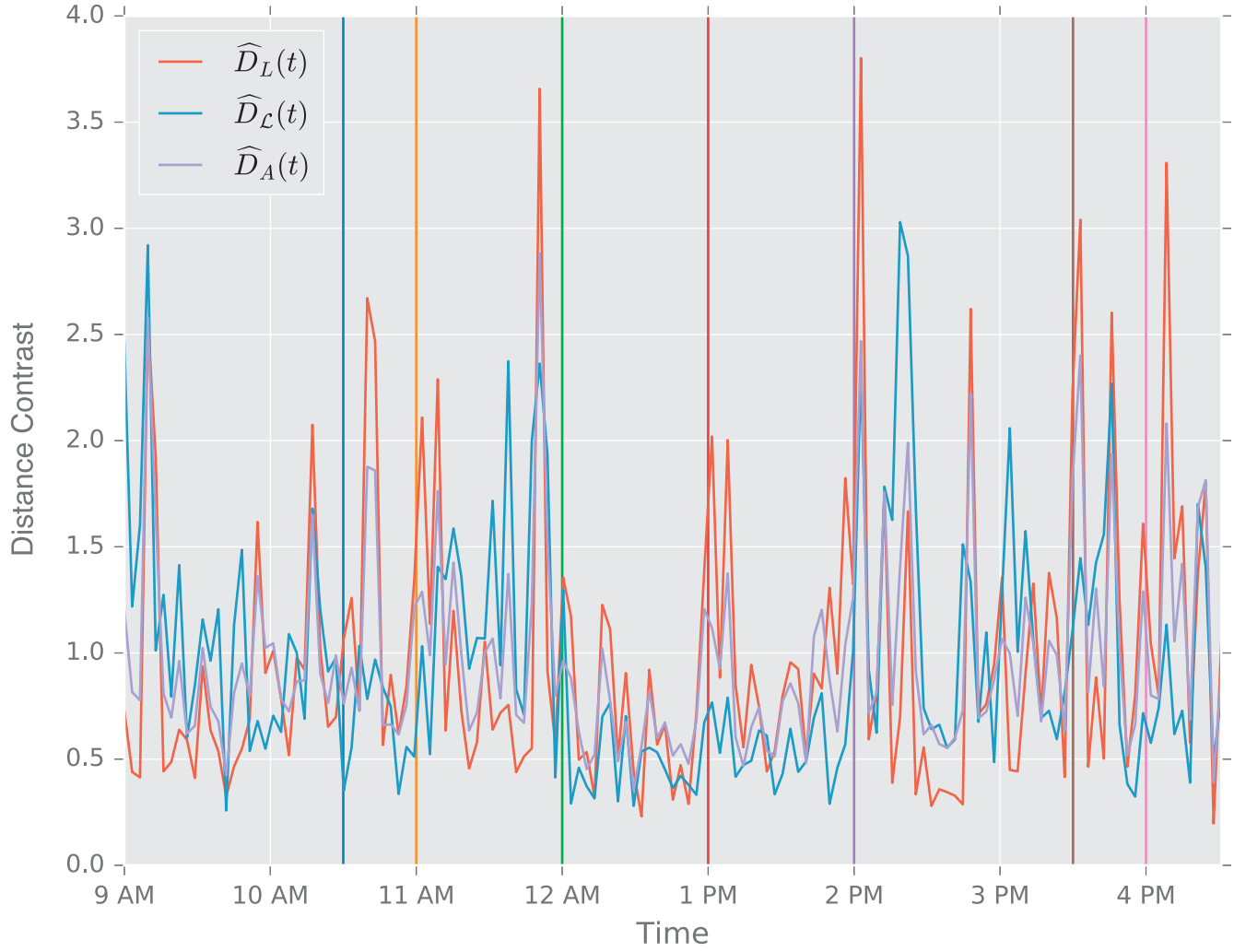

Fig 14. Primary school data set: Normalized temporal differences for the three spectral distances: Combinatorial Laplacian $\hat{D}_L$, normalized Laplacian $\hat{D}_{\mathcal{L}}$, and adjacency $\hat{D}_A$.

https://doi.org/10.1371/journal.pone.0228728.g014

strength of the functional coupling between brain regions) connectomes, and the comparison of unweighted (we only record if two regions are functionally coupled—irrespective of the strength of that coupling) connectomes. Furthermore, for each type of connectome, we can vary the density of edges (and therefore the volume) by varying the threshold used to define functional connectivity. We used two values of the threshold for the Pearson correlation coefficient: 0.5 and 0.8.

Figs 19 and 20 display the distance contrast between the set of unweighted ASD connectomes and the set of control connectomes for two values of the threshold: 0.5 and 0.8 respectively. Figs 21 and 22 display the distance contrast between the weighted ASD and control connectomes for two values of the threshold: 0.5 and 0.8 respectively.

Unfortunately, no distances can effectively separate the two ensembles of connectomes. Indeed, the negative median of the distance contrast $\hat{D}_1$ indicates that the distance between two connectomes in the ASD and control populations respectively, D_1, is on average *lower* than the average distances between two connectomes from the control population, μ_0 (see (12)).

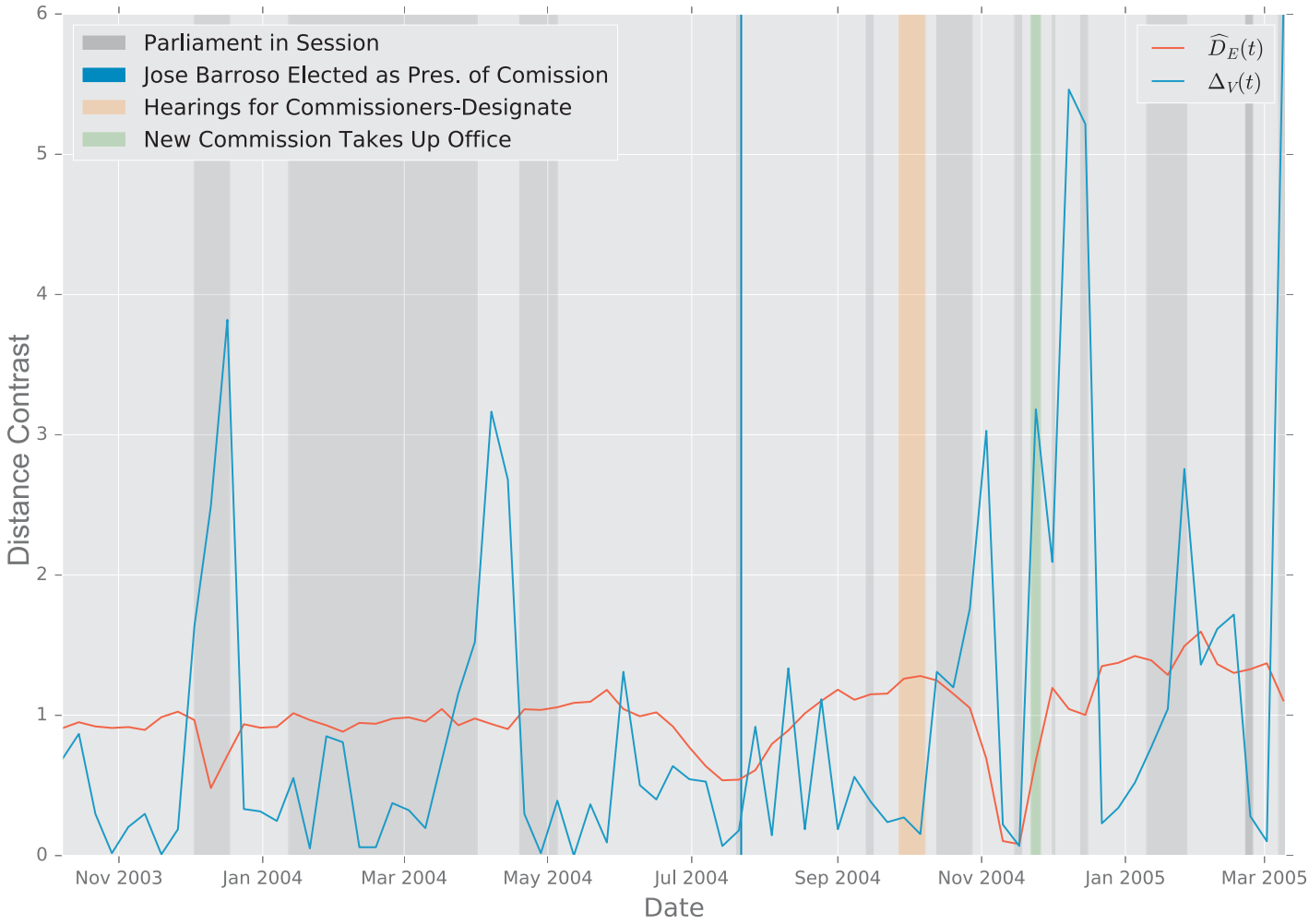

Fig 15. EU-emails: Normalized temporal difference, for the edit distance $\hat{D}_E$, and absolute value of the changes in the graph volume over time.

https://doi.org/10.1371/journal.pone.0228728.g015

This result indicates that the variability in the control population is greater than the contrast between the two populations. A refined analysis, provided in Section 4.2.3, shows that the structural differences between the two graph ensembles are localized within subsets of edges, and cannot be detected when one compares both complete sets of edges. Furthermore, the local changes in connectivity are of the same order of magnitude as the random local variations present in these connectomes. For these two reasons, a global comparison using graph metrics seems ineffective for this problem.

Table 5. Events related to the activity of the European Parliament during 2004 [106].

Date	Event
22 July	Jose Barroso elected as President of the Commission
27 September–8 October	Hearings of the Commissioners-designates, nominated by Jose Barroso
22–26 November	New Commission takes up office

https://doi.org/10.1371/journal.pone.0228728.t005

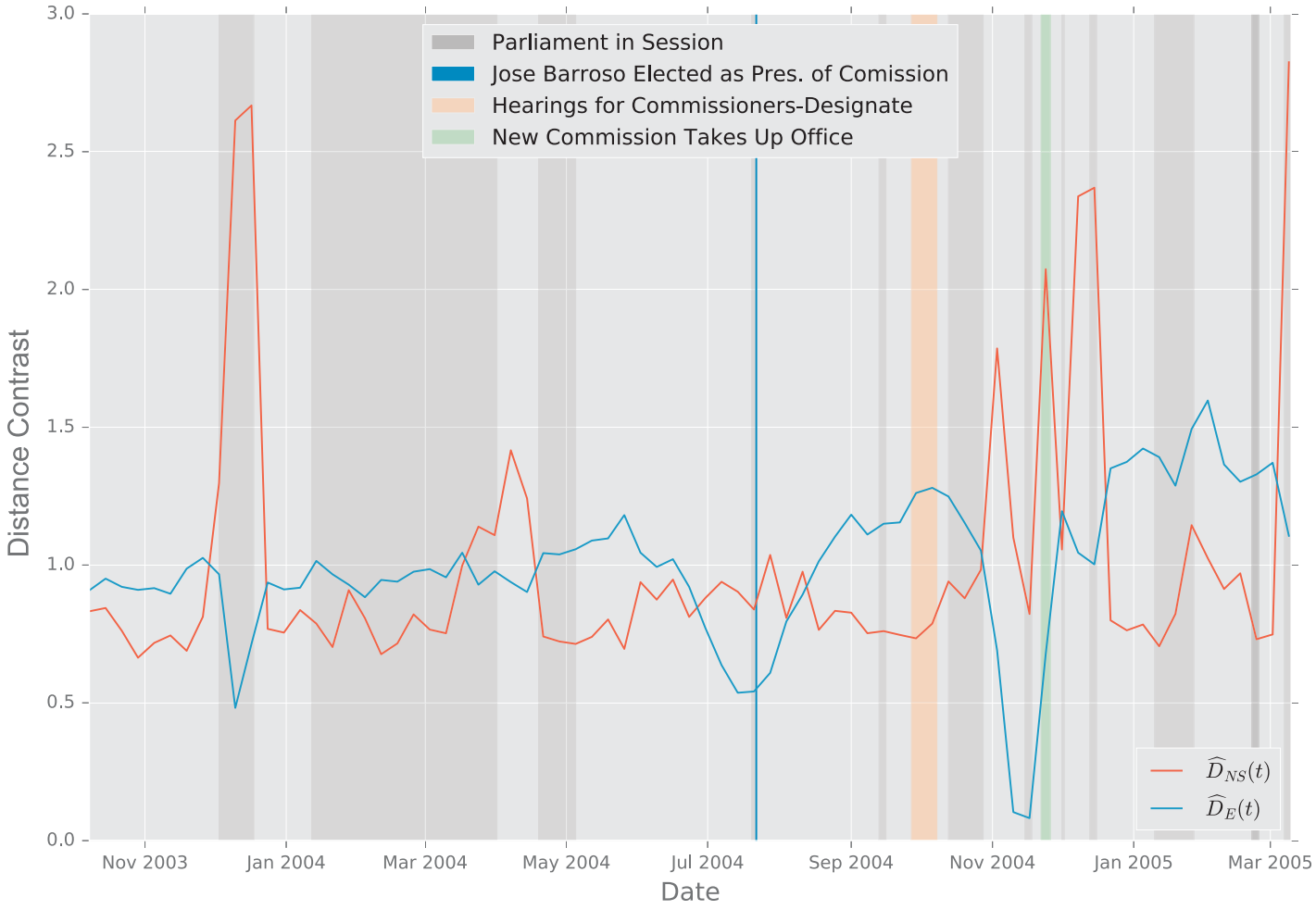

Fig 16. EU-emails: Normalized temporal differences for the NetSimile distance $\hat{D}_{NS}$ and edit distance $\hat{D}_{E}$.

https://doi.org/10.1371/journal.pone.0228728.g016

4 Discussion

This section provides an analysis of the numerical simulations and the results of experiments conducted on real world graphs. Because the numerical simulations were performed using random graph ensembles, we provide some theoretical justification to explain our findings. The performance of the distances is studied using a "multiscale lens": we organize distances according to the scale at which they aptly detect changes within a graph. We consider three classes of scales: (1) the fine scale of the local connectivity, formed by the ego-net; (3) the very large scale associated with communities; and finally (2) a mesoscale that bridges the scales from the local to the global scales. Interestingly, this multiscale paradigm has inspired methods to synthesize networks with guaranteed structural properties at multiple different scales [107].

For all the graph ensembles studied in this work, we expect random fine scale changes triggered by the stochastic nature of the models. At the other end of the scale, we expect that certain changes in connectivity may have dramatic large scale changes. Fig 23 provides a cartoon of this concept: changes in edge connectivity of a stochastic blockmodel (left) result in the creation of a preferential attachment model with the prototypical presence of hubs with very high degree.

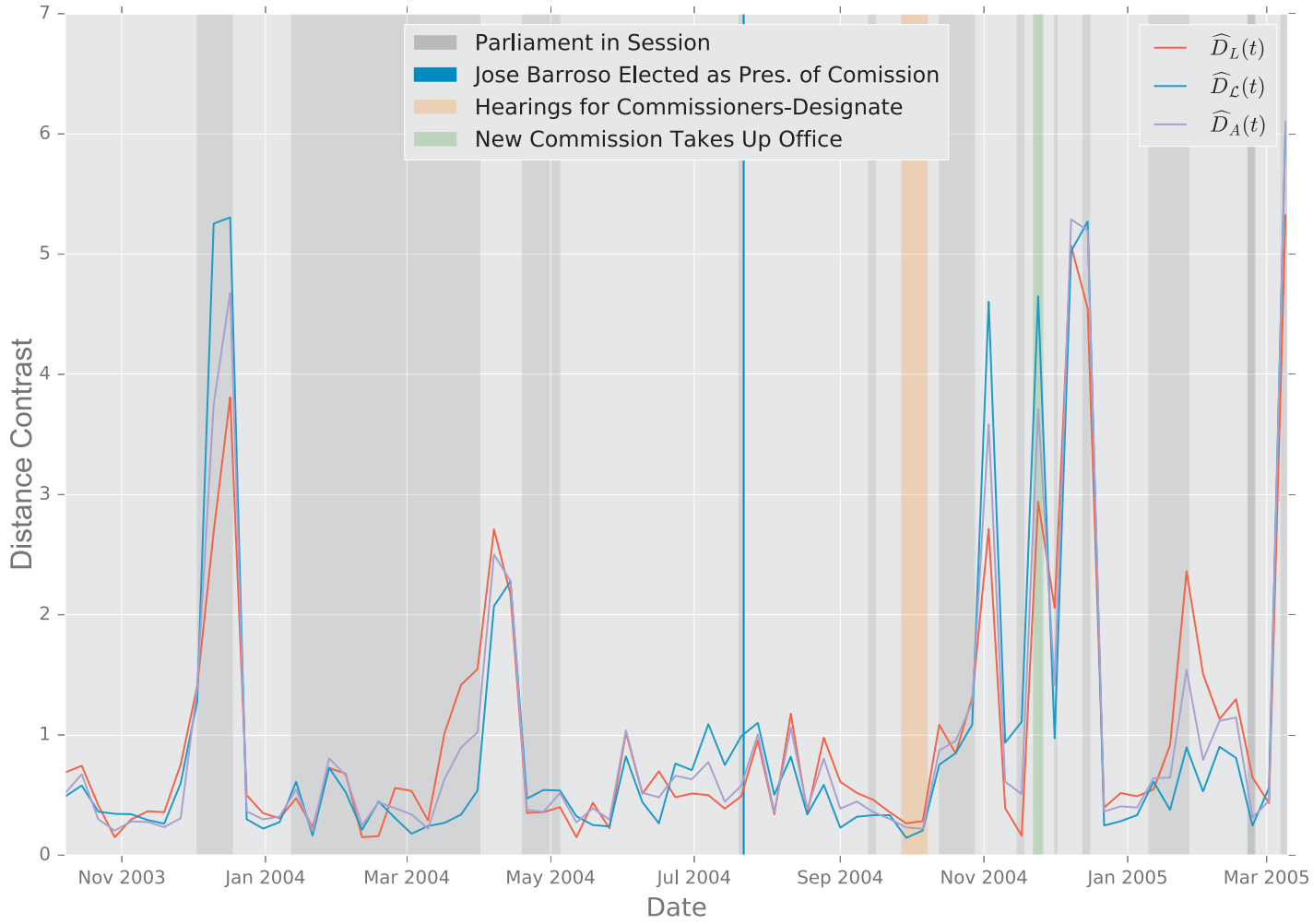

Fig 17. EU-emails: Normalized temporal differences for the three spectral distances: Combinatorial Laplacian $\hat{D}_L$, normalized Laplacian $\hat{D}_{\mathcal{L}}$, and adjacency $\hat{D}_A$.

https://doi.org/10.1371/journal.pone.0228728.g017

Finally, the power-law degree distribution of the preferential attachment model suggests that the graph connectivity involves multiple scales spanning from the finest scale up to the coarsest scale. Distances adapted to these "mesoscales" should be optimal to detect these graphs. Similarly, we expect that the small world model require the analysis of connectivity at the mesoscale.

4.1 The multiscale detection of random graph ensembles

4.1.1 Detecting large scale changes. In this study, we focus on experiments where the coarse-scale structure involves the presence (or absence) of communities. The prototypical ensemble to study the ability of distances to detect communities is the stochastic blockmodel.

We study a stochastic blockmodel with two partitions of equal size, and we thus expect the second eigenvalue λ_2 (of either one of the three matrices) to be the primary distinguishing spectral feature of the graph. Fig 4 confirms our analysis. We conjecture that k eigenvalues be needed to detect k communities. Indeed, the authors in [85] have shown that the spectrum of the adjacency matrix is composed of two distinct components. A continuous spectrum (the bulk) that is centered around 0 is a modified version of the classic semicircle law. The discrete

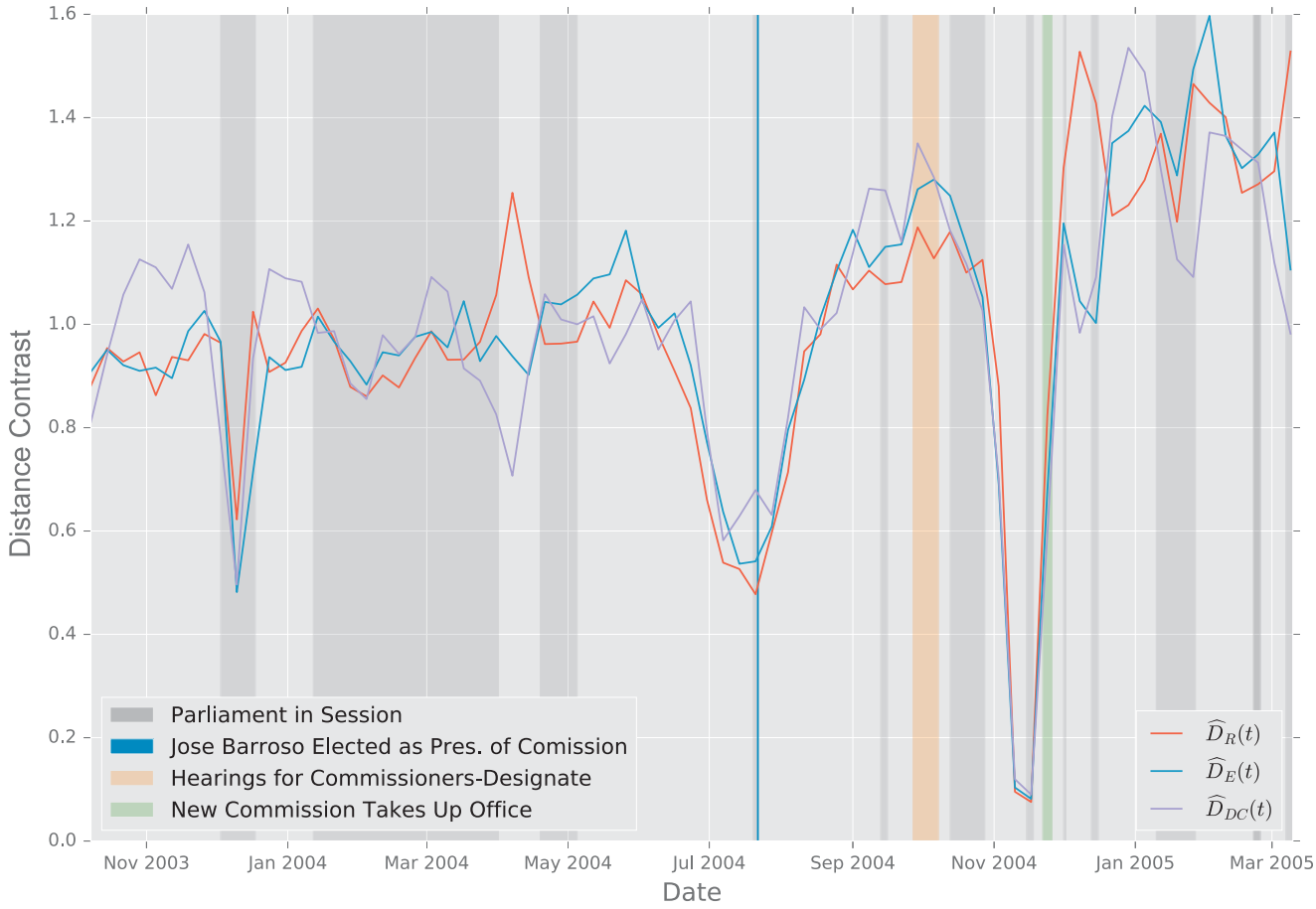

Fig 18. EU-emails: Normalized temporal differences for the resistance distance $\hat{D}_R$, edit distance $\hat{D}_E$, and DeltaCon distance $\hat{D}_{DC}$.

https://doi.org/10.1371/journal.pone.0228728.g018

spectrum is the second component; it is composed of discrete eigenvalues, distributed away from the continuous spectrum. The number of discrete eigenvalues is equal in number to the number of communities in the network. The separation between the continuous and the discrete spectra is what allows our spectral distances to function effectively in detecting community structure.

Fig 24 shows the empirical spectral densities of the adjacency matrices ($\lambda^{\boldsymbol{A}}$) for the stochastic blockmodel (blue) and the uncorrelated random graph (orange). The density are well separated around the second largest eigenvalue $\lambda_2^{\boldsymbol{A}}$. The bulk of the spectra for both models overlap significantly, and provide no hope of separating the models from these eigenvalues. Consequently, using additional eigenvalues decreases the contrast by including noise in the comparison (see Fig 4).

The use of the spectrum for community partitioning in graphs has a long history (e.g., [108] and references therein). Recently, Lee et al. [15] have proven a performance bound on the effectiveness of using the first k eigenvectors to partition the graph into k clusters. In practice, if the graph includes more than two communities of different sizes, the optimal contrast will require more than the first non trivial eigenvalue.

In summary, we find that when examining global structure, the adjacency spectral distance and DeltaCon distance both provide good performance. When examining community structure in particular, one need not employ the full spectrum when using a spectral distance.

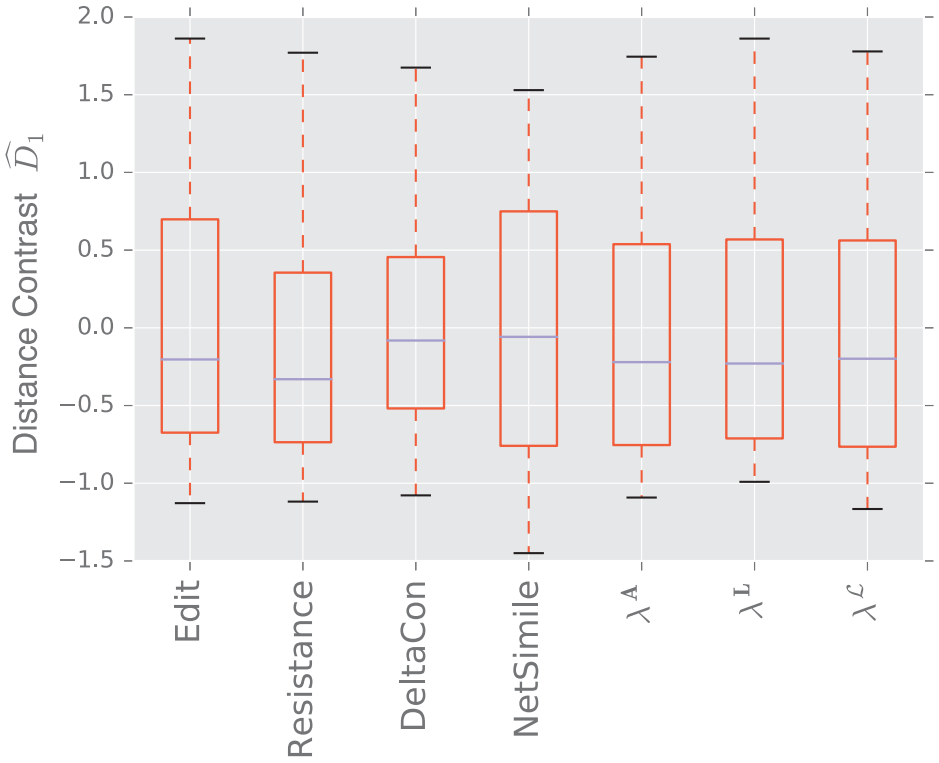

Fig 19. ABIDE data set: Distance contrast $\hat{D}_1$ between the unweighted ASD and control connectomes for a threshold $T = 0.5$.

https://doi.org/10.1371/journal.pone.0228728.g019

4.1.2 Detecting mesoscale changes. In this paper we studied two random graph ensembles whose connectivity structures span several scales: (1) the preferential attachment model with a non-negligible number of highly connected vertices (hubs) and a large number of vertices with low degree; (2) the Watts-Strogatz model where high-degree vertices are extremely unlikely, and where generative rewiring mechanism does not result in the presence of communities in the graph.

To differentiate graphs based on mesoscale connectivity structures, one should use a spectral distance computed from either the combinatorial graph Laplacian or the adjacency matrix.

The Combinatorial Graph Laplacian Spectral Distance. We find that the best tool for detecting graphs whose degree distribution exhibits polynomial decay [7] is the combinatorial Laplacian spectral distance. The presence of the degree matrix D in the Laplacian $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ means that comparison of Laplacians is very effective for discerning between models with radically different degree distributions. Since significant differences between the degree distributions of the preferential and attachment graphs occur in the tail (i.e. high-degree vertices), the inclusion of the final few eigenvalues is essential if one wishes to use the Laplacian spectrum to perform this comparison.

Fig 25 displays the empirical spectral densities of the normalized Laplacian ($\lambda^{\boldsymbol{L}}$) for the preferential attachment model (blue) and the uncorrelated random graph (orange). We observe qualitatively, as demonstrated in [19], that the tails of the Laplacian spectrum of a preferential attachment graph exhibits polynomial decay similar to the tail of the degree distribution. This is a prime example of the way in which the spectrum of the Laplacian can be heavily influenced by the degree distribution.

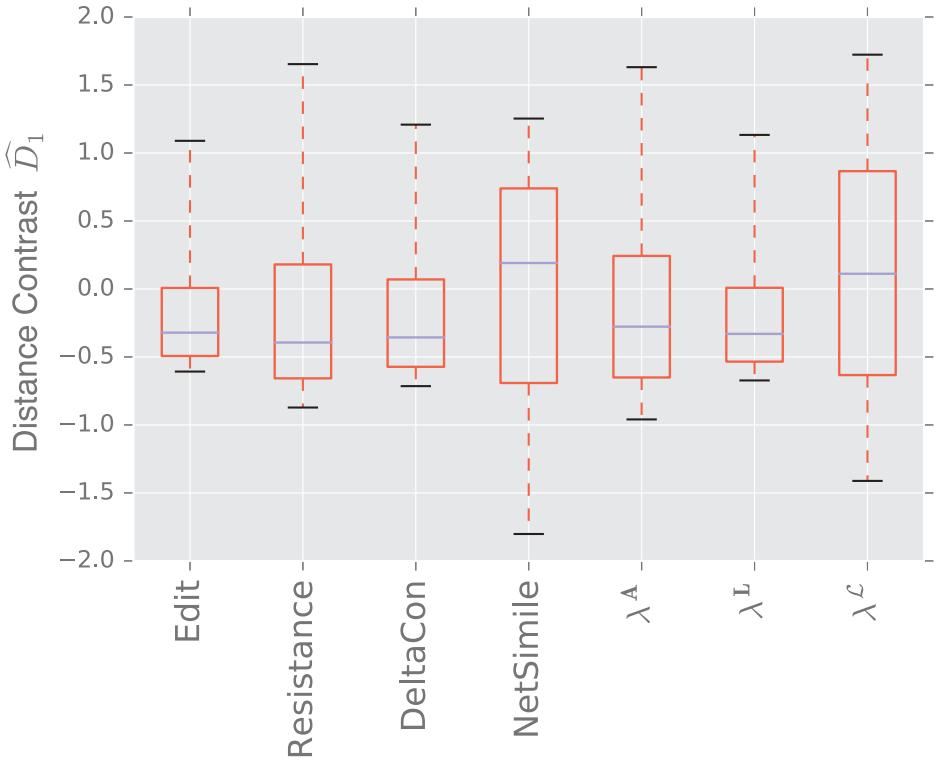

Fig 20. ABIDE data set: Distance contrast $\hat{D}_1$ between the unweighted ASD and control connectomes for a threshold $T = 0.8$.

https://doi.org/10.1371/journal.pone.0228728.g020

The Adjacency Spectral Distance. Our findings indicate that the adjacency spectral distance is the optimal distance for detecting graphs with short average distances, such as the Watts-Strogatz. Farkas et al. [19] argue that the presence of a high number of triangles is the distinguishing feature of a Watts-Strogatz model. The third moment of the spectral density of $\boldsymbol{A}$ yields the expected number of triangles in a graph, and so one would expect inclusion of the full spectrum important in detecting the topological signature of this model.

This theoretical analysis is confirmed in our experiments. We see in Fig 9 that inclusion of the large-k (high frequency) eigenvalues is essential to differentiate between the Watts-Strogatz and the random graph models. Fig 26 confirms that the empirical spectral density of the Watts-Strogatz model exhibits high skewness, requiring the inclusion of the bulk of the spectrum to be able to differentiate this model from the random uncorrelated graph (see Fig 26.

A more refined analysis confirms that the very fine scale connectivity, such as the degree distribution, of the Watts-Strogatz is similar to that of the random graph model, and therefore the inclusion of the high modes (high $\lambda_k^{\boldsymbol{A}}$) decreases the contrast between the two models (see Fig 9).

4.1.3 Impact of local structure

Is the local structure signal or noise? In this work we consider that the local scale is defined by the local connectivity at the level of each vertex. Because our study relies on random graphs ensembles, the local connectivity is intrinsically random. In some of the graph models, the generative model that leads to the realization of the random graphs induce some coupling

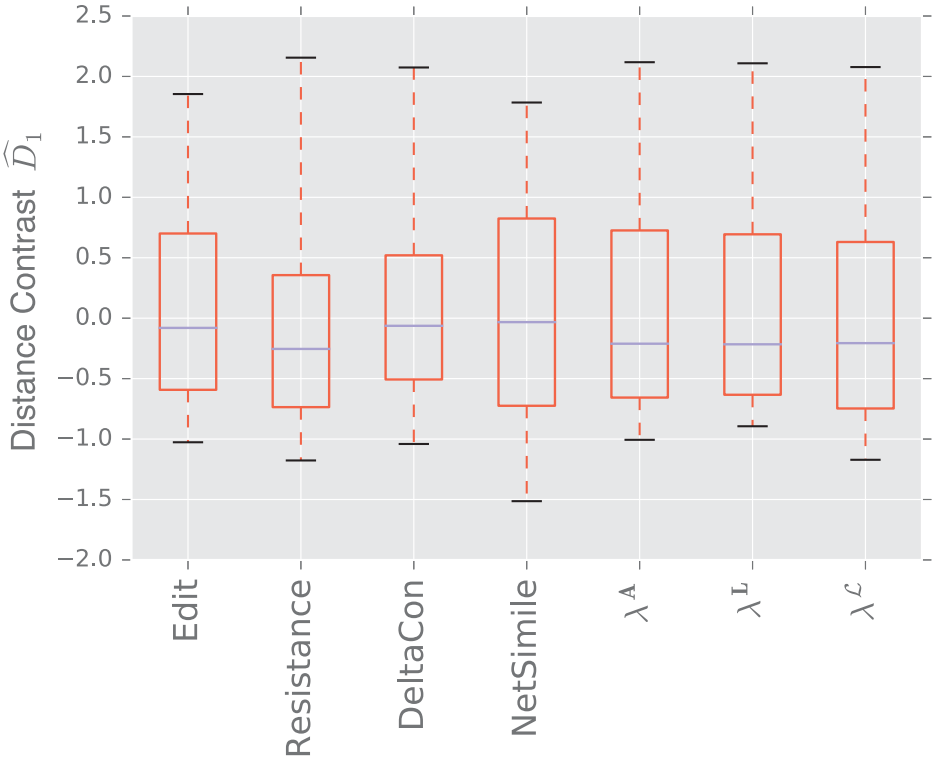

Fig 21. ABIDE data set: Distance contrast $\hat{D}_1$ between the weighted ASD and control connectomes for a threshold $T = 0.5$.

https://doi.org/10.1371/journal.pone.0228728.g021

across the scales. The fine scale statistics, such as the degree distribution, become a "window" on larger scale patterns of connectivity that happen at multiple scales.

We provide two examples of such phenomena. We first revisit the preferential attachment model (see Section 3.1.2). Fig 6 shows that the spectral distance based on the combinatorial graph Laplacian needs the very fine scale, or high frequencies, provided by the eigenvalues $\lambda_k^{\boldsymbol{L}}$ for large k to detect the preferential attachment model. This is interesting, since this local scale is determined by the degree distribution. Should the null model mimic the degree distribution of the preferential attachment model (see Section 3.1.3) then the two graphs become indistinguishable (see Fig 7). In this example, the fine scale clearly provides a "signature" of the graph connectivity.

The lattice graph is another extreme example of where the local connectivity structure can be used to identify the graph. The lattice graph includes cycles of any size (starting with length 4). As a result, the spectral distances all benefit from increasing the number of eigenvalues used to compare the graphs (see Fig 11).

On the other hand, random fluctuations can also be a source of uninformative noise when comparing graphs. The results of Section 3.1.1 illustrate this fact. The Laplacian spectrum is unable to distinguish between the stochastic blockmodel and the uncorrelated random graph, while the normalized Laplacian distinguishes them well. The difference between these two matrix representations is that normalization removes degree information, which is not informative in this particular model (see Fig 4).

A similar problem arise when we apply the resistance distance to the stochastic blockmodel; as discussed in the previous section, the resistance distance is disproportionately influenced by local structure, and is unable to discern the global structure of the graph over local fluctuations.

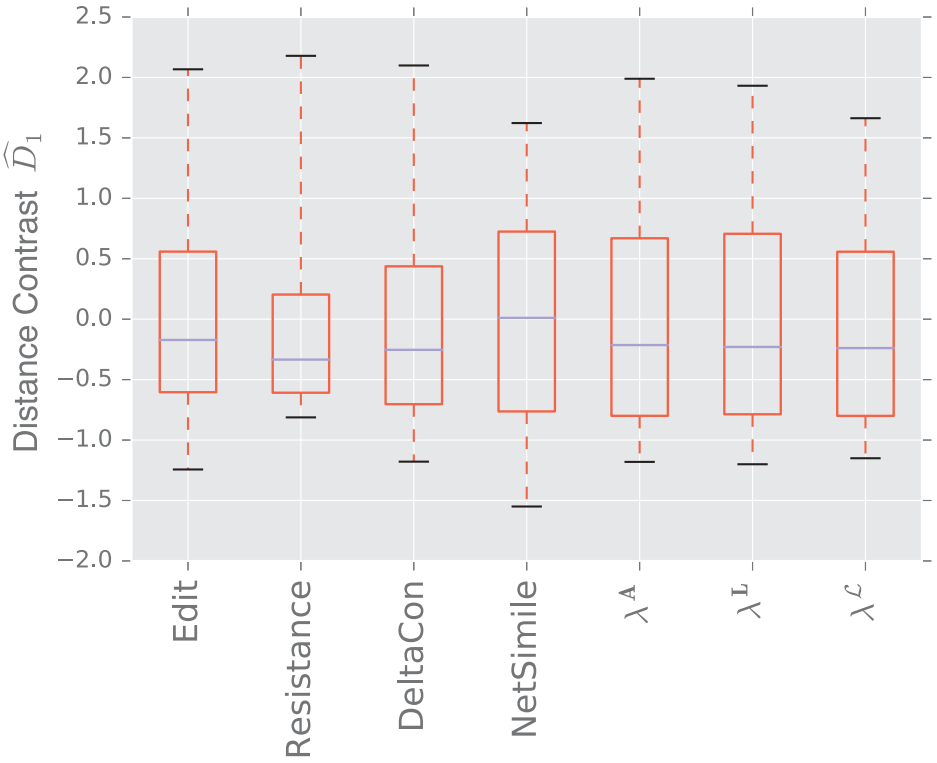

Fig 22. ABIDE data set: Distance contrast $\hat{D}_1$ between the weighted ASD and control connectomes for a threshold $T = 0.8$.

https://doi.org/10.1371/journal.pone.0228728.g022

Interestingly, DeltaCon does not appear to suffer from local fluctuations as much as the resistance distance. This could be due to the structure of the matrix **S** that DeltaCon uses to represent the graph, or due to the use of the Matusita distance rather than the ℓ_1 or ℓ_2 norm to compare the resulting matrices (for more discussion of this, see Sections 2.2 and 3.1 in [30]).

In summary, it is essential to determine whether local topological features are of interest in the comparison problem at hand; inclusion of locally targeted distance measures can hinder the performance of graph distances in cases where local structure is noisy and uninformative. However, if local structure is ignored, one can often omit essential structural information about the graphs under comparison.

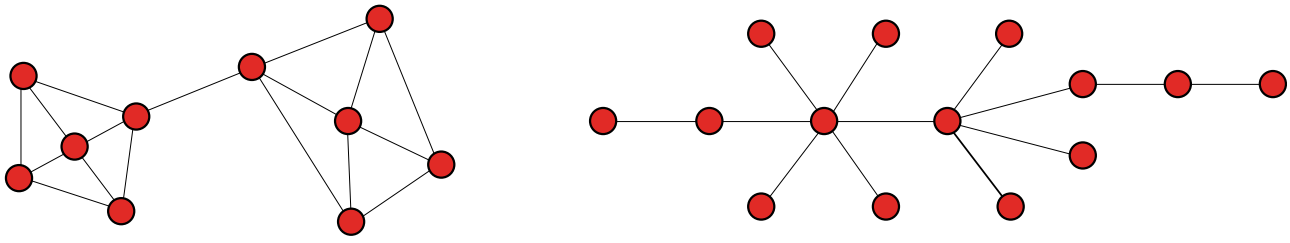

Fig 23. Two significant global structures observed in our experiments. On the left is the community structure typical of the stochastic blockmodel. On the right is the heavy-tailed degree distribution typical of the preferential attachment model.

https://doi.org/10.1371/journal.pone.0228728.g023

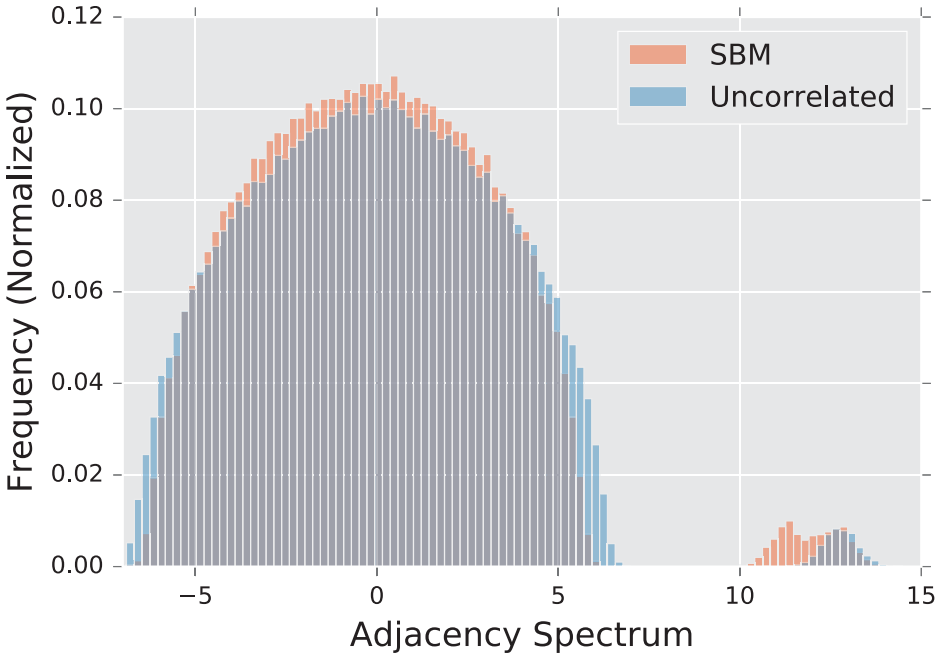

Fig 24. Empirical spectral densities λ^A of the adjacency matrix for the stochastic blockmodel (blue) and the uncorrelated random graph (orange). Densities are built from an ensemble of 1,000 graphs generated using parameters described in section 3.1.1.

https://doi.org/10.1371/journal.pone.0228728.g024

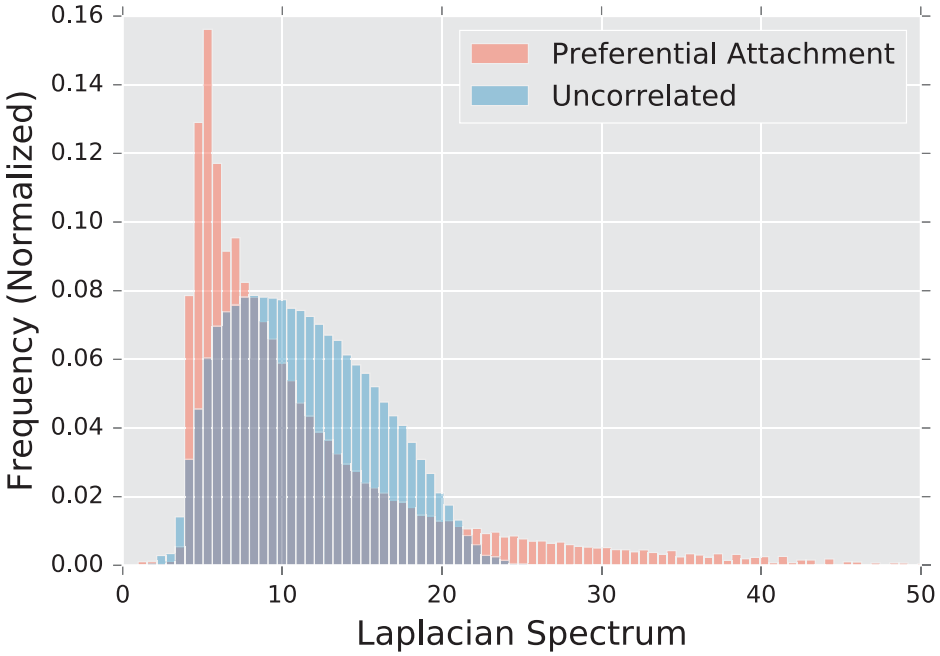

Fig 25. Empirical spectral densities λ^L of the combinatorial Laplacian for the preferential attachment model (blue) and the uncorrelated random graph (orange). Densities are built from an ensemble of 1,000 graphs generated using parameters described in section 3.1.2.

https://doi.org/10.1371/journal.pone.0228728.g025

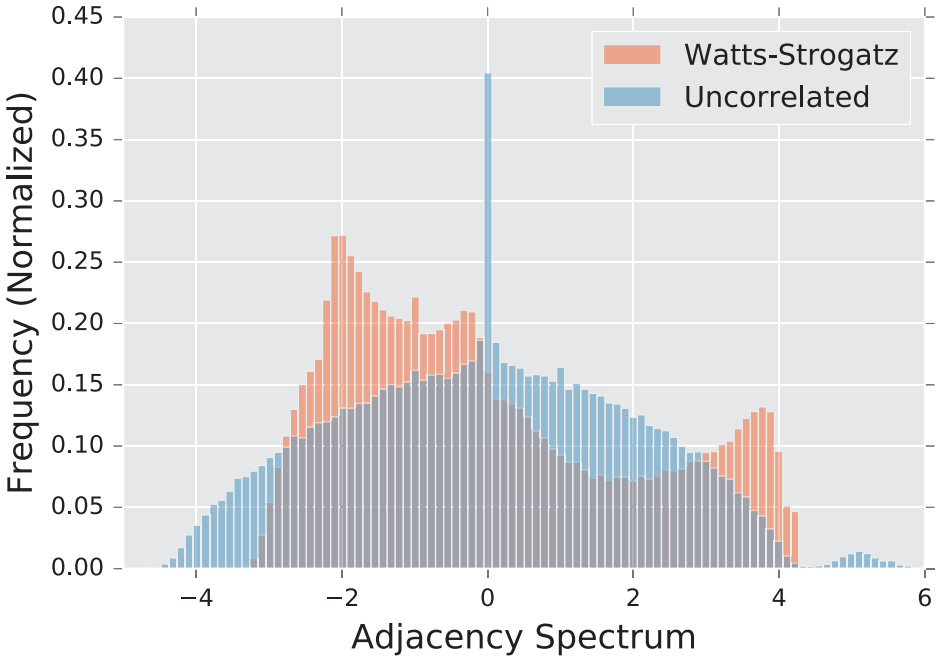

Fig 26. Empirical spectral densities λ^A of the adjacency matrix for the Watts-Strogatz model (blue) and the uncorrelated random graph (orange). Densities are built from an ensemble of 1,000 graphs generated using parameters described in section 3.1.4. The uncorrelated random graph model has a value of p that is smaller than those used in the previous models, creating a sharp peak at $\lambda^A = 0$.

https://doi.org/10.1371/journal.pone.0228728.g026

4.2 Real world networks

Experiments performed on random graph ensembles provide a mechanism to gauge the ability of each distance to detect changes in structural features that are prototypical of the corresponding ensembles (e.g., communities, clustering coefficients, power law degree distribution, etc.). Specifically, this analysis lends itself to a systematic exploration of an experimental version of the two-sample test problem where we compare two populations of random graphs using a distance statistic, and we experimentally test whether both populations could be generated by the same probability distribution.

In this context, we explore the two-sample test problem in neuroscience, and compared two populations of functional brain networks. Signal-to-noise is a ubiquitous problem in analyzing actual graph data, and is particularly notable in building connectivity networks of human brain activity (see e.g., [109]). Accordingly, the results of our data experiments show that in the presence of real-world noise levels, many of these distances fail to distinguish subtle structural differences. In the face of this, we examine more targeted analysis techniques that may be applied in such a situation.

A related question concerns the change point detection scenario for a dynamic graph, where we detect significant changes between adjacent time steps using a distance [4]. The first experiment suggests that the tools that perform the most consistently in the two-sample test problem (the spectral distances) are unreliable in the change detection experiment. This experiment is interesting because it allows us to evaluate distances in a context where graphs exhibit significant volume fluctuations, a situation that we did not encounter in our numerical studies.

4.2.1 Primary school face to face contact. The primary school face to face contact dataset (3.2.1) provides a real-world example to evaluate the performance of distances in the context of a dynamic network.

The purpose of the analysis is to assess whether distances can detect changes in the topology coupled with the hidden events that control the network topology and connectivity (such as those that occur during the lunch period). We are also interested to verify if distances are robust against random changes within each classroom that do not affect the communication between the classes (e.g., see Fig 2 at times 10:50 a.m., 10:57 a.m.).

The most remarkable conclusion of this particular experiment is that although the spectral distances are very efficient and stable for the purposes of comparing two random graphs sampled from distinct probability models (see section 3.1), these distances perform poorly in the context of change point detection (see Fig 14). In contrast, the resistance distance can detect subtle topological changes that are coupled to latent events that dynamically modify the networks. The resistance distance remains impervious to random local changes, which do not affect the large scale connectivity structure (see Fig 12).

Unlike the analysis of random graphs, where the volume of the two graphs were always the same, the volume of the dynamic network changes rapidly, and therefore the edit distance exhibits significant changes throughout the school day. While the edit distance can reliably monitor large scale changes in the graph volume, it entirely misses the significant events that disrupt the graph topology: onset and end of morning recess, onset of first lunch, end of second lunch (see Fig 13).

With the help of Fig 2 (the snapshots are obtained from the movie available on [110]), we analyze some of the most significant differences between the three distances.

At time 10:20 a.m., $\hat{D}_R$ changes abruptly (see Fig 12) as a result of a massive increase in the number of contacts between students in the second, third, and fourth grades (see Fig 2). While $\hat{D}_E$ and $\hat{D}_{DC}$ also register this change, they are less sensitive to the merging of the communities than the resistance-perturbation distance.

The significant difference between $\hat{D}_R$ and the two other distances before 11:00 a.m. (see Fig 12) is also very interesting. Because the recess period is winding down from 10:50 a.m. to 10:57 a.m., the number of contacts within each class decreases very significantly (especially in the two second grade classes, see Fig 2). These *within the classes* changes are easily detected by $\hat{D}_{DC}$ and $\hat{D}_E$, which continue to grow during this time interval. However, there are only very few changes in the contacts *between the classes* during that period (see Fig 2). Consequently, $\hat{D}_R$ becomes very small (see the significant dip of $\hat{D}_R$ shortly before 11:00 a.m. in Fig 12).

The resistance-perturbation distance $\hat{D}_R$ is also able to detect the dissolution of the classes at 11:57 a.m. just before the official lunch period, as the students are running outside of the classrooms into the hallway. The random appearance of the connectivity (see Fig 2) reflects the activity of the students. This is significant, because this event happens before the number of contacts increases (the edit distance jumps right after 12:p.m., see Fig (see Fig 13).

The geometry of the graph at 12:13 p.m. (see Fig 2) is indicative of the fact that half of the students take their lunch in the cafeteria, while the other half play in the courtyard. In spite of the fact that the DeltaCon distance is still large, the contact network is in fact in a large scale stable topological configuration, leading to a small $\hat{D}_R$.

In fact, $\hat{D}_R$ can also detect the pattern of activity associated with the second lunch period at 12:54 p.m. (see Fig 12). Because this is only a reconfiguration of the network, the edit distance is oblivious to these changes (see Fig 13).

Finally, we note that the resistance distance can detect the early regrouping of the students around 1:46 p.m. (see Fig 12), according to their classroom (see Fig 2), before the end of the lunch period.

Interestingly, $\hat{D}_R$ can also detect the small number of students who are late going back to their class between 2:00 p.m. and 2:03 p.m. (see Fig 2). The edit and DeltaCon distances are not affected by the removal of these cross-community edges (see Figs 13, 12 respectively), but the resistance distance is greatly influenced by these changes, and thus $\hat{D}_R$ is very large (see Fig 12).

These structural changes are of a global nature. In Sections 3.1.1 and 4.1.1 we saw that the spectral distances were more effective than the matrix distances to detect large-scale differences between the following two graph ensembles: the stochastic blockmodel and the uncorrelated random graph model. Because the dynamic network is comprised of communities, a naive analysis would suggest that spectral distances should also outperform the resistance perturbation distance and DeltaCon.

A refined analysis demonstrates that if a dynamic network is composed of communities, the resistance perturbation provides the ideal solution to the change point detection problem, by effectively ignoring the rapid random fluctuations at the node level, while remaining sensitive to changes in connectivity between communities.

This analysis requires that we review the expression of the resistance-perturbation in terms of the eigenvalues and eigenvectors of the normalized graph Laplacian [9]. The present authors derived in [9], a closed-form expression of the resistance-perturbation distance between a graph and a rank-one perturbation of that graph, wherein a single edge has changed. This theoretical analysis is useful indeed, because it provides the baseline scenario to compare various changes in connectivity in the context of the change point detection scenario for a dynamic graph.

We recall that ϕ_k denotes the k^{th} eigenvector of the normalized graph Laplacian, with the eigenvalues organized as $0 = \lambda_1^{\mathcal{L}} \leq \ldots \leq \lambda_n^{\mathcal{L}}$. We also denote by $\mathrm{d}_{\mathrm{rp}\,1}(G, G + \Delta w_{i_0 j_0})$ the resistance perturbation distances between a graph G, and the graph obtained from G by a perturbation $\Delta w_{i_0 j_0}$ to the edge (i_0, j_0), $G + \Delta w_{i_0 j_0}$.

Theorem 1 (resistance-perturbation after edge modification [9]) *If $G + \Delta w_{i_0 j_0}$ is the graph obtained from G by a perturbation $\Delta w_{i_0 j_0}$ to the edge (i_0, j_0), then*

$$\mathrm{d}_{\mathrm{rp}\,1}(G, G + \Delta w_{i_0 j_0}) = \frac{2n|\Delta w_{i_0 j_0}|}{1 + \Delta w_{i_0 j_0} R_{i_0 j_0}} \sum_{k=2}^{n} \frac{1}{(\lambda_k^{\mathcal{L}})^2} [\phi_k(i_0) - \phi_k(j_0)]^2. \tag{15}$$

As expected, the term

$$\frac{\Delta w_{i_0 j_0}}{1 + \Delta w_{i_0 j_0} R_{i_0 j_0}}, \tag{16}$$

controls the size of the resistance-perturbation distance $\mathrm{d}_{\mathrm{rp}\,1}(G, G + \Delta w_{i_0 j_0})$.

More interestingly, the sum $\sum_{k=2}^{n} [\phi_k(i_0) - \phi_k(j_0)]^2 / (\lambda_k^{\mathcal{L}})^2$ in (15) provides the "frequency response" of the graph to the perturbation. This response can be analyzed as follows. For small k, the eigenvalues $\lambda_k^{\mathcal{L}}$ are small, and the corresponding eigenvectors ϕ_k "oscillate" very slowly on the graph, i.e. $\phi_k(i_0) - \phi_k(j_0) \approx 0$ unless i_0 and j_0 belong to different nodal regions. In this latter case, the effect of the edge perturbation $\Delta w_{i_0 j_0}$ will be maximal. An example of this phenomenon corresponds to the primary school face-to-face contact network, where each classroom forms a densely connected community. The classrooms are weakly connected to one another. For the same $\Delta w_{i_0 j_0}$, $\mathrm{d}_{\mathrm{rp}\,1}(G, G + \Delta w_{i_0 j_0})$ will be maximal if i_0 and j_0 are in different classrooms, and will be very small if the two nodes belong to the same classroom.

For large k, eigenvectors ϕ_k "oscillate" very quickly on the graph, making it difficult to estimate the contribution of $[\phi_k(i_0) - \phi_k(j_0)]^2$. This issue is mitigated by the fact that the weights $1/(\lambda_k^{\mathcal{L}})^2$ are relatively small, since the eigenvalues $\lambda_k^{\mathcal{L}}$ are large.

In summary, the resistance-perturbation provides a multiscale analysis that automatically de-emphasize the random variability at the very fine scales, to wit the distance "denoises" the graph dynamic (see also [111, 112] for similar analyses).

We note the existence of topological distance [113–115] that also provide "multiscale distances" through a filtration process. These topological distances go beyond the scope of the current study that focuses on geometric distances.

4.2.2 European Union Emails. The analysis of the pattern of connectivity of email exchanges between the members of a European research institute provides a second example of dynamic network. The first observation is that the rate of changes in the number of emails (volume of each graphs) appears to be influenced by the calendar of the European Parliament (see Fig 15). As suspected by the authors in [105], the activity in this dataset appears to be influenced by a series of events at the European Parliament in Brussels and Strasbourg. This conjecture is confirmed by the analysis of the spectral distances (see Fig 17): the three distances show large values whenever the session of the Parliament is resumed (e.g., 3 December, 2003; 19 April 2004, 13 October 2004, etc.) or adjourned (e.g., 18 December 2003, 1 April 2004, 28 October 2004, etc.) More interestingly, all the distances become suddenly very large during the entry into office of the new 2004-2009 European Commission (22-26 November, 2004).

Unlike all the spectral distances, the resistance distance and the DeltaCon distance are able to detect the election of Jose Barroso as President of the European Commission (22 July 2004) (see Fig 18). We had already observed and analysed the performance of these two distances in the context of the face-to-face contact networks. This experiment confirms that the resistance perturbation provides a very sensitive statistic to detect changes in dynamic networks.

4.2.3 Functional brain connectivity. The experiments in section 3.2.3 leave the choice of the global distance open. In this section, we gain further insight into the analysis of this dataset, and demonstrate that local changes in the connectivity of functional brain networks can indeed be detected. These minute changes require assigning a more robust weight along the edges, and designing a distance between graphs that can be tuned to respond to local changes at specific scales (a weighted spectral distance). The implementation of these ideas is beyond the scope of the paper.

In order to gain some understanding into the inability of the graph distance to differentiate between the ASD patients and the controls, we revisit the original data, and compute the following contrast for each pair of nodes (i, j) in the network

$$\hat{D}_{i,j} \stackrel{\text{def}}{=} \frac{\hat{\mathbb{E}}[\rho_{i,j}^1] - \hat{\mathbb{E}}[\rho_{i,j}^0]}{\sigma(\rho_{i,j}^0)}, \tag{17}$$

where $\hat{\mathbb{E}}[\rho_{i,j}^1]$ is the sample mean correlation between regions i and region j of the brain atlas, computed over all ASD subjects (population 1). Similarly, $\hat{\mathbb{E}}[\rho_{i,j}^0]$ and $\sigma(\rho_{i,j}^0)$ are the sample mean and sampled variance, respectively, of the correlations between regions i and region j of the brain atlas, computed over all controls (population 0). Fig 27 displays the contrast $\hat{D}_{i,j}$ for all pair of regions in the AAL atlas, $i, j = 1, \ldots 116$.

To ease the interpretation of this matrix, consider for instance a value of $\hat{D}_{i,j} = 0.25$. This value indicates that (on average) the functional correlation $\hat{\mathbb{E}}[\rho_{i,j}^1]$ between regions i and j of

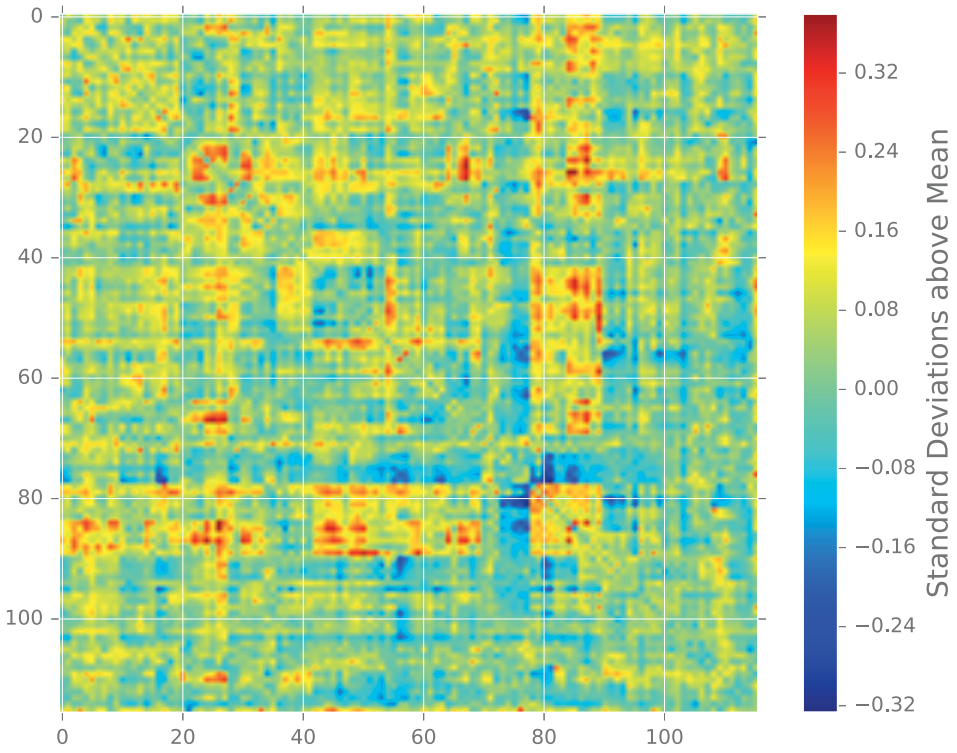

Fig 27. Contrast $\hat{D}_{i,j}$ for each pair of regions (i, j) in the atlas AAL computed between the ASD subjects and the controls (see (17)).

https://doi.org/10.1371/journal.pone.0228728.g027

ASD subjects is above the (average) correlation of the controls, $\hat{\mathbb{E}}[\rho^1_{i,j}]$, by 0.25× the standard deviation of the correlation (of the controls), $\sigma(\rho^0_{i,j})$.

A visual inspection of Fig 27 highlights the presence of localized high contrast between the two populations. A closer examination reveals that ASD subjects are underconnected in regions 73 through 77, and are overconnected in regions 79 and 84 through 89. Table 6 provides a list of regions that are show anomalous functional connectivity.

Table 6. Regions with anomalous connectivity. Correspondence between labels and regions is established via the Automated Anatomical Labelling atlas [116].

Label	Region	Connection
73	L. Putamen	Underconnected
74	R. Putamen	Underconnected
75	L. Globus Pallidus	Underconnected
76	R. Globus Pallidus	Underconnected
77	L. Thalamus	Underconnected
79	R. Transverse Temporal Gyrus	Overconnected
84	R. Superior Temporal Lobe	Overconnected
85	L. Middle Temporal Gyrus	Overconnected
86	R. Middle Temporal Gyrus	Overconnected
87	L. Middle Temporal Pole	Overconnected
88	R. Middle Temporal Pole	Overconnected
89	L. Inferior Temporal Gyrus	Overconnected

https://doi.org/10.1371/journal.pone.0228728.t006

Fig 27 confirms that there exist structural differences between the connectomes of ASD subjects and controls. Unfortunately, our analysis shows that these differences are smaller than one standard deviation of the correlation of the controls $\sigma(\rho^0_{i,j})$ (see the color bar in Fig 27). To further aggravate this situation, we note that the pattern of anomalous connectivity are isolated, while the vast majority of correlations are very close to zero (green cells in Fig 27). We conclude that the low amplitude of the contrast $\hat{D}_{i,j}$ and its sparsity contribute to our inability to use graph distances to detect significant changes between the connectomes of the two populations (see also [117] for a detailed analysis of regional connectivity). We note that others have reported similar findings [118, 119].

5 Conclusion

The success of statistical machine learning relies on the construction of sophisticated spaces of signals (functional spaces) wherein properties of algorithms can be rigorously evaluated. The core of the analysis usually relies on the existence of bases that reveal the properties of the class of functions of interest. There currently is no equivalent for the study of graph ensembles.

In this paper, we considered existing ensembles of random graphs as prototypical examples of certain graph *structures*, which are the building blocks of existing real world networks. These ensembles were used to rigorously analyze various graph distances in the context of the two-sample test problem.

Specifically, we studied the ability of various distances to compare two samples randomly drawn from distinct ensembles of graphs. We investigated the relationship between the families of graph ensembles, the structural features characteristic of these ensembles, and the sensitivity of the distances to these characteristic structural features. The performance of the distances is studied using a "multiscale lens": we organize distances according to the scale at which they aptly detect changes within a graph. We consider three classes of scales: (1) the fine scale of the local connectivity, formed by the ego-net; (3) the very large scale associated with communities; and finally (2) a mesoscale that bridges the scales from the local to the global scales.

We concluded our study with experiments conducted on real-world networks, where we study the two-sample test problem for networks of functional brain connectivity, and we detected change points in a dynamic network of face-to-face contacts.

5.1 Recommendations

Throughout this study, we observed that the adjacency spectral distance (see Sections 3.1 and 3.2) exhibits good performance across a variety of scenarios, making it a reliable choice for a wide range of problems. Spectral distances also exhibit practical advantages over matrix distances, as they can inherently compare graphs of different sizes and can compare graphs without known vertex correspondence. The adjacency spectrum in particular is well-understood, and is perhaps the most frequently studied graph spectrum; see e.g., [19, 83]. Finally, fast, stable eigensolvers for symmetric matrices are ubiquitous in modern computing packages such as ARPACK, NumPy, and Matlab, allowing for rapid deployment of models based on spectral graph comparison. The Python library `NetComp` [75] further simplifies the application of these tools to practical problems; see Section 7 for more details. Furthermore, randomized algorithms for matrix decomposition allow for highly parallelizable calculation of the spectra of large graphs [73].

However, the utility of the adjacency spectral distance is not general enough to simply apply it to any given two-sample test or anomaly detection problem in a naive manner. A prudent practitioner would combine exploratory structural analysis of the graphs in question with

an ensemble approach in which multiple distance measures are considered simultaneously, and the resulting information is combined to form a consensus. Such systems are commonplace in problems of classification in machine learning, where they are sometimes known as "voting classifiers" (see e.g., [120]).

In this study, we have been comparing graphs of equal volume (in expectation). In situations where the graph volume varies drastically (e.g., see Section 2.5.1), the process of choosing a graph comparison tool may differ significantly. The situation reverses when we look at the problem of detecting change points in a dynamic graph (see Section 2.5.1). In this scenario, the matrix distances proved most effective in detecting changes in the latent variables controlling the network dynamics. The spectral distances, on the other hand, were so noisy as to be useless. When trying to detect change points in a dynamic graph, one computes the distance between consecutive time steps. In this scenario the two graphs being compared share many more edges than in the two-sample test. As demonstrated in section 4.2.1, the resistance perturbation distance, or DeltaCon, yield exemplary performance. We note that we found that raw fluctuations in graph volume did not yield useful information about the latent processes that triggered changes in graph connectivity.

Based on the results of our experiments, we provide a suggested decision process in Fig 28. If the graphs to be compared exhibit differences in volume or size, then these should be examined to see if they hold predictive power, as they are simple and efficient to compute. If they prove ineffective, then one must consider the setting. In a dynamic setting, in which a dynamic graph is being compared at subsequent time steps, then we recommend using matrix distances based on the results of Section 2.5.1. If one is comparing graphs to determine whether a sample belongs to a given population, then the adjacency spectral distance is the most reliable, as Sections 3.1 and 3.2 demonstrate. Finally, if none of these approaches give adequate performance, then a more targeted analysis must be performed, such as the edge-wise statistical comparison of weights in Fig 27. The particular design of this analysis is domain specific and highly dependent upon the nature of the data.

6 Notation

For reference, in Table 7 we provide a table of notation used throughout the paper.

7 NetComp: Network comparison in python

`NetComp` is a Python library that implements the graph distances studied in this work. Although many useful tools for network construction and analysis are available in the well-known `NetworkX` [78], advanced algorithms such as spectral comparisons and DeltaCon are not present. `NetComp` is designed to bridge this gap.

7.1 Design consideration

The guiding principles behind the library are

1. **Speed**. The library implements algorithms that run in linear or near-linear time, and are thus applicable to large graph data problems. See below regarding the implementation of exact and approximate forms of DeltaCon and the resistance distance.
2. **Flexibility**. The library uses as its fundamental object the adjacency matrix. This matrix can be represented in either a dense (`NumPy` matrix) or sparse (`SciPy` sparse matrix) format. Using such a ubiquitous format as fundamental allows easy input of graph data from a wide variety of sources.

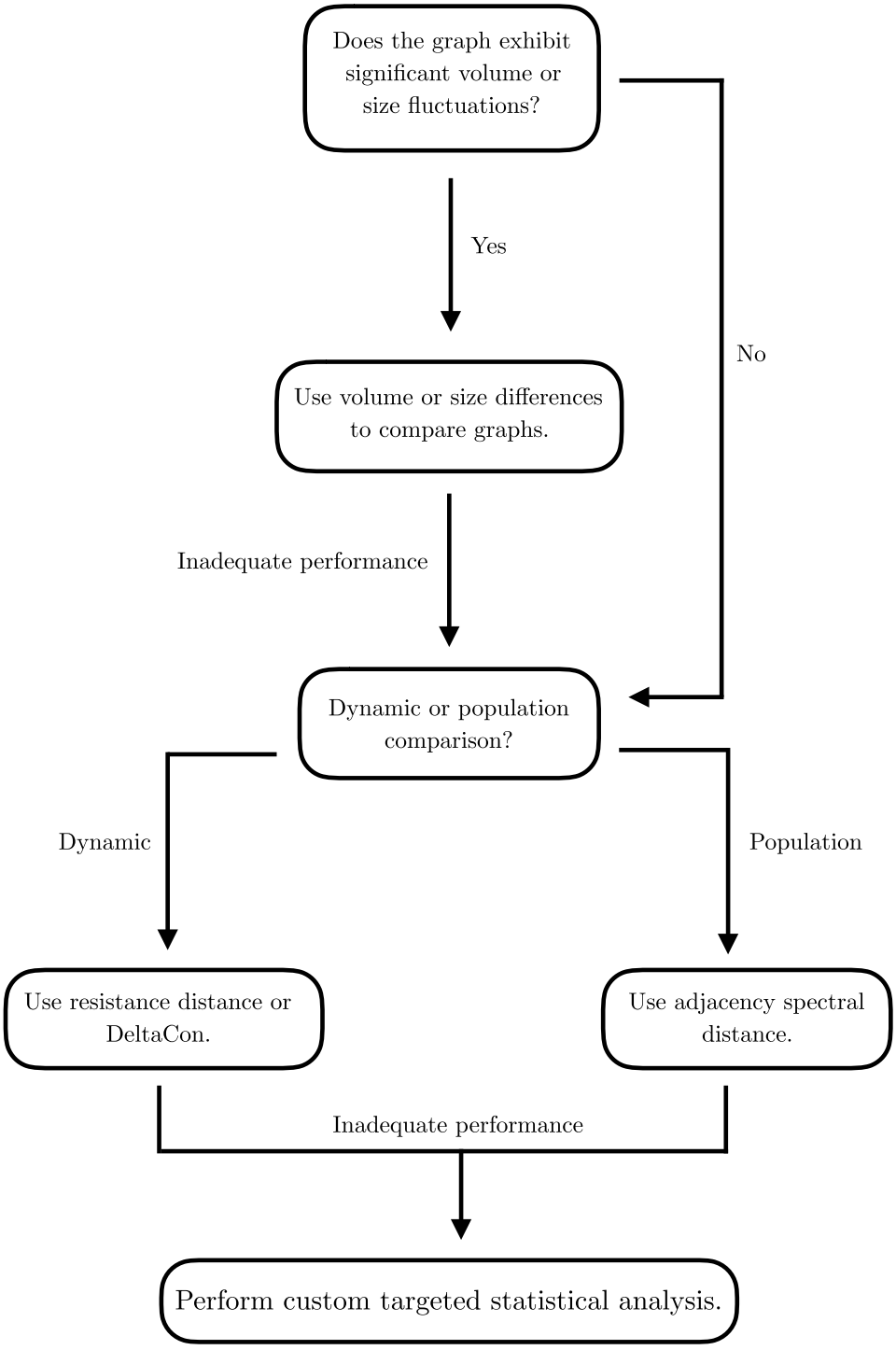

Fig 28. Flow chart summarizing the suggested decision process for applying distance measures in empirical data.

https://doi.org/10.1371/journal.pone.0228728.g028

Table 7. Table of commonly used notation.

G	Graph
V	Vertex set, taken to be $\{1, 2, \ldots, n\}$
E	Edge set, subset of $V \times V$
W	Weight function, $W : E \to \mathbb{R}^+$
n	Size of the graph, $n = \lvert V \rvert$
m	Number of edges, $m = \lvert E \rvert$
d_i	Degree of vertex i
$\boldsymbol{D}$	Degree matrix (diagonal)
$d(\cdot, \cdot)$	Distance function
$\boldsymbol{A}$	Adjacency matrix
$\boldsymbol{L}$	Laplacian matrix
$\mathcal{L}$	normalized Laplacian matrix (symmetric)
λ_i^A	i^{th} eigenvalue of the adjacency matrix
λ_i^L	i^{th} eigenvalue of the Laplacian matrix
$\lambda_i^{\mathcal{L}}$	i^{th} eigenvalue of the normalized Laplacian matrix
$\mathcal{G}_{\{0,1\}}$	The {null,alternative} population of graphs
$G_{\{0,1\}}$	Sample graph from $\mathcal{G}_{\{0,1\}}$
$\mathcal{D}_0$	Distribution of distances between graphs in null population
D_0	Sample from $\mathcal{D}_0$
$\mathcal{D}_1$	Distribution of distances $d(G_0, G_1)$
D_1	Sample from $\mathcal{D}_1$
$\hat{\mathcal{D}}_1$	Distribution of the contrast $\mathcal{D}_1$ between D_0 and D_1, (see (12))
$\hat{D}_1$	Sample from $\hat{\mathcal{D}}_1$
$G(n, p)$	Uncorrelated random graph with parameters n and p
(n, p, q)	Parameters for stochastic blockmodel
(n, l)	Parameters for preferential attachment model, with $1 < l \leq n$
(n, k)	Parameters for Watts-Strogatz graph, with $k < n$ even

https://doi.org/10.1371/journal.pone.0228728.t007

3. **Extensibility**. The library is written so as to be easily extended by anyone wishing to do so. The included graph distances will hopefully be only the beginning of a full library of efficient modern graph comparison tools that will be implemented within `NetComp`.

`NetComp` is available via the Python Package Index, that is most frequently accessed via the command-line tool `pip`. The user can install it locally via the shell command

```
pip install netcomp.
```

As of writing, the library is in alpha. The approximate (near-linear) forms of DeltaCon and the resistance distance are not yet included in the package. Both algorithms have an quadratic-time exact form that is implemented. Those interested can download the source code and contribute (by adding the distance of their choice) at https://www.github.com/peterewills/netcomp.

Acknowledgments

The authors are grateful to the anonymous reviewers and the Academic Editor for their insightful comments and suggestions that greatly improved the content and presentation of this manuscript.

Author Contributions

Conceptualization: Peter Wills, François G. Meyer.

Formal analysis: Peter Wills, François G. Meyer.

Investigation: Peter Wills, François G. Meyer.

Methodology: Peter Wills, François G. Meyer.

Software: Peter Wills.

Supervision: François G. Meyer.

Writing – original draft: Peter Wills.

Writing – review & editing: Peter Wills, François G. Meyer.

References

1. Cook D JH LB, editor. Mining Graph Data. Wiley; 2006.
2. Haslbeck JM, Waldorp LJ. How well do network models predict observations? On the importance of predictability in network models. Behavior research methods. 2018; 50(2):853–861. https://doi.org/10.3758/s13428-017-0910-x PMID: 28718088
3. Liu W, Chen PY, Yu F, Suzumura T, Hu G. Learning Graph Topological Features via GAN. IEEE Access. 2019; 7:21834–21843. https://doi.org/10.1109/ACCESS.2019.2898693
4. Holme P, Saramäki J. Temporal networks. Physics Reports. 2012; 519(3):97–125. https://doi.org/10.1016/j.physrep.2012.03.001.
5. Donnat C, Holmes S. Tracking network dynamics: A survey using graph distances. The Annals of Applied Statistics. 2018; 12(2):971–1012. https://doi.org/10.1214/18-AOAS1176
6. Akoglu L, Tong H, Koutra D. Graph based anomaly detection and description: a survey. Data Mining and Knowledge Discovery. 2015; 29(3):626–688. https://doi.org/10.1007/s10618-014-0365-y
7. Barabási AL, Albert R. Emergence of scaling in random networks. Science. 1999; 286(5439):509–512. https://doi.org/10.1126/science.286.5439.509 PMID: 10521342
8. Watts DJ, Strogatz SH. Collective dynamics of 'small-world' networks. Nature. 1998; 393(6684):440–442. https://doi.org/10.1038/30918 PMID: 9623998
9. Monnig ND, Meyer FG. The resistance perturbation distance: A metric for the analysis of dynamic networks. Discrete Applied Mathematics. 2018; 236:347–386. https://doi.org/10.1016/j.dam.2017.10.007.
10. Erdős P, Rényi A. On Random Graphs I. Publicationes Mathematicae. 1959; 6:290–297.
11. Wilson RC, Zhu P. A study of graph spectra for comparing graphs and trees. Pattern Recognition. 2008; 41(9):2833–2841. https://doi.org/10.1016/j.patcog.2008.03.011.
12. Biyikoglu T, Leydold J, Stadler PF. Laplacian eigenvectors of graphs: Perron-Frobenius and Faber-Krahn type theorems. Springer; 2007.
13. Rudin W. Functional Analysis. International series in pure and applied mathematics. McGraw-Hill; 1991. Available from: https://books.google.com/books?id=Sh_vAAAAMAAJ.
14. de Lange S, de Reus M, Van Den Heuvel M. The Laplacian spectrum of neural networks. Frontiers in computational neuroscience. 2014; 7:189. https://doi.org/10.3389/fncom.2013.00189 PMID: 24454286
15. Lee JR, Gharan SO, Trevisan L. Multiway Spectral Partitioning and Higher-Order Cheeger Inequalities. J ACM. 2014; 61(6):37:1–37:30. https://doi.org/10.1145/2665063
16. Haemers WH, Spence E. Enumeration of cospectral graphs. European Journal of Combinatorics. 2004; 25(2):199–211. https://doi.org/10.1016/S0195-6698(03)00100-8
17. Schwenk AJ. Almost all trees are cospectral. New directions in the theory of graphs. 1973; p. 275–307.
18. Bhamidi S, Evans SN, Sen A. Spectra of Large Random Trees. Journal of Theoretical Probability. 2012; 25(3):613–654. https://doi.org/10.1007/s10959-011-0360-9
19. Farkas IJ. Spectra of "real-world" graphs: Beyond the semicircle law. Physical Review E. 2001; 64(2). https://doi.org/10.1103/PhysRevE.64.026704

20. Wigner EP. On the Distribution of the Roots of Certain Symmetric Matrices. The Annals of Mathematics. 1958; 67(2):325–327. https://doi.org/10.2307/1970008
21. O'Rourke S. Gaussian Fluctuations of Eigenvalues in Wigner Random Matrices. Journal of Statistical Physics. 2010; 138(6):1045–1066. https://doi.org/10.1007/s10955-009-9906-y
22. Chung FRK. Spectral Graph Theory. American Mathematical Society; 1997.
23. Friedman J, Tillich JP. Wave equations for graphs and the edge-based Laplacian. Pacific Journal of Mathematics. 2004; 216(2):229–266. https://doi.org/10.2140/pjm.2004.216.229
24. Maas C. Computing and interpreting the adjacency spectrum of traffic networks. Journal of Computational and Applied Mathematics. 1985; 12-13(Supplement C):459–466. https://doi.org/10.1016/0377-0427(85)90039-1.
25. Moore EF. The shortest path through a maze. Proceedings of an International Symposium on the Theory of Switching. 1959; p. 285–292.
26. Ellens W, Spieksma FM, Mieghem PV, Jamakovic A, Kooij RE. Effective graph resistance. Linear Algebra and its Applications. 2011; 435(10):2491–2506. https://doi.org/10.1016/j.laa.2011.02.024.
27. Haveliwala TH. Topic-sensitive PageRank: a context-sensitive ranking algorithm for Web search. IEEE Transactions on Knowledge and Data Engineering. 2003; 15(4):784–796. https://doi.org/10.1109/TKDE.2003.1208999
28. Goddard W, Oellermann OR. Distance in Graphs. In: Structural Analysis of Complex Networks. Birkhäuser Boston; 2011. p. 49–72. Available from: https://doi.org/10.1007/978-0-8176-4789-6_3.
29. Wills P, Meyer FG. Change Point Detection in a Dynamic Stochastic Blockmodel. In: Complex Networks and Their Applications VIII. Springer; 2020. p. 211–222.
30. Koutra D, Shah N, Vogelstein JT, Gallagher B, Faloutsos C. Delta Con: Principled Massive-Graph Similarity Function with Attribution. ACM Transactions on Knowledge Discovery from Data (TKDD). 2016; 10(3):28. https://doi.org/10.1145/2824443
31. Koutra D, Ke TY, Kang U, Chau DHP, Pao HKK, Faloutsos C. Unifying Guilt-by-Association Approaches: Theorems and Fast Algorithms. In: Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg; 2011. p. 245–260.
32. Berlingerio M, Koutra D, Eliassi-Rad T, Faloutsos C. NetSimile: A Scalable Approach to Size-Independent Network Similarity. CoRR. 2012;abs/1209.2684.
33. Bassett DS, Bullmore E, Verchinski BA, Mattay VS, Weinberger DR, Meyer-Lindenberg A. Hierarchical organization of human cortical networks in health and schizophrenia. Journal of Neuroscience. 2008; 28(37):9239–9248. https://doi.org/10.1523/JNEUROSCI.1929-08.2008 PMID: 18784304
34. Kaiser M. A tutorial in connectome analysis: topological and spatial features of brain networks. Neuroimage. 2011; 57(3):892–907. https://doi.org/10.1016/j.neuroimage.2011.05.025 PMID: 21605688
35. van den Heuvel MP, Sporns O, Collin G, Scheewe T, Mandl RCW, Cahn W, et al. Abnormal rich club organization and functional brain dynamics in schizophrenia. JAMA Psychiatry. 2013; 70(8):783–792. https://doi.org/10.1001/jamapsychiatry.2013.1328 PMID: 23739835
36. Goyal P, Ferrara E. Graph embedding techniques, applications, and performance: A survey. Knowledge-Based Systems. 2018; 151:78–94. https://doi.org/10.1016/j.knosys.2018.03.022
37. Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G. The graph neural network model. IEEE Transactions on Neural Networks. 2008; 20(1):61–80. https://doi.org/10.1109/TNN.2008.2005605 PMID: 19068426
38. Yanardag P, Vishwanathan S. Deep graph kernels. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM; 2015. p. 1365–1374.
39. Li Y, Gu C, Dullien T, Vinyals O, Kohli P. Graph Matching Networks for Learning the Similarity of Graph Structured Objects. arXiv preprint arXiv:190412787. 2019.
40. Coley CW, Barzilay R, Green WH, Jaakkola TS, Jensen KF. Convolutional embedding of attributed molecular graphs for physical property prediction. Journal of chemical information and modeling. 2017; 57(8):1757–1772. https://doi.org/10.1021/acs.jcim.6b00601 PMID: 28696688
41. Fout A, Byrd J, Shariat B, Ben-Hur A. Protein interface prediction using graph convolutional networks. In: Advances in Neural Information Processing Systems; 2017. p. 6530–6539.
42. Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. Neural message passing for quantum chemistry. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org; 2017. p. 1263–1272.
43. Ktena SI, Parisot S, Ferrante E, Rajchl M, Lee M, Glocker B, et al. Distance metric learning using graph convolutional networks: Application to functional brain networks. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer; 2017. p. 469–477.

44. Preuer K, Renz P, Unterthiner T, Hochreiter S, Klambauer G. Fréchet ChemNet distance: a metric for generative models for molecules in drug discovery. Journal of chemical information and modeling. 2018; 58(9):1736–1741. https://doi.org/10.1021/acs.jcim.8b00234 PMID: 30118593

45. Yang K, Swanson K, Jin W, Coley C, Eiden P, Gao H, et al. Analyzing Learned Molecular Representations for Property Prediction. Journal of Chemical Information and Modeling. 2019; 59(8):3370–3388. https://doi.org/10.1021/acs.jcim.9b00237 PMID: 31361484

46. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:160902907. 2016.

47. Li Y, Tarlow D, Brockschmidt M, Zemel R. Gated graph sequence neural networks. arXiv preprint arXiv:151105493. 2015.

48. Monti F, Boscaini D, Masci J, Rodolà E, Svoboda J, Bronstein MM. Geometric deep learning on graphs and manifolds using mixture model CNNs. arXiv preprint arXiv:161108402. 2016.

49. Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G. The graph neural network model. IEEE Transactions on Neural Networks. 2009; 20(1):61–80. https://doi.org/10.1109/TNN.2008.2005605 PMID: 19068426

50. Bruna J, Zaremba W, Szlam A, LeCun Y. Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:13126203. 2013.

51. Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in neural information processing systems; 2016. p. 3844–3852.

52. Battaglia PW, Hamrick JB, Bapst V, Sanchez-Gonzalez A, Zambaldi V, Malinowski M, et al. Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:180601261. 2018.

53. Zhou J, Cui G, Zhang Z, Yang C, Liu Z, Sun M. Graph neural networks: A review of methods and applications. arXiv preprint arXiv:181208434. 2018.

54. Kriege NM, Johansson FD, Morris C. A Survey on Graph Kernels. arXiv preprint arXiv:190311835. 2019.

55. Xu K, Hu W, Leskovec J, Jegelka S. How powerful are graph neural networks? arXiv preprint arXiv:181000826. 2018.

56. Morris C, Ritzert M, Fey M, Hamilton WL, Lenssen JE, Rattan G, et al. Weisfeiler and Leman go neural: Higher-order graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33; 2019. p. 4602–4609.

57. Oneto L, Navarin N, Donini M, Sperduti A, Aiolli F, Anguita D. Measuring the expressivity of graph kernels through statistical learning theory. Neurocomputing. 2017; 268:4–16. https://doi.org/10.1016/j.neucom.2017.02.088

58. Gromov M, Katz M, Pansu P, Semmes S. Metric structures for Riemannian and non-Riemannian spaces. Birkhauser; 2007.

59. Berger M. A Panoramic View of Riemannian Geometry. Springer Verlag; 2003.

60. Villani C. Optimal transport: old and new. vol. 338. Springer; 2008.

61. Mémoli F, Sapiro G. A theoretical and computational framework for isometry invariant recognition of point cloud data. Foundations of Computational Mathematics. 2005; 5(3):313–347. https://doi.org/10.1007/s10208-004-0145-y

62. Mémoli F. Gromov–Wasserstein distances and the metric approach to object matching. Foundations of computational mathematics. 2011; 11(4):417–487.

63. Panaretos VM, Zemel Y. Statistical aspects of Wasserstein distances. Annual review of statistics and its application. 2019; 6:405–431. https://doi.org/10.1146/annurev-statistics-030718-104938

64. Peyré G, Cuturi M. Computational optimal transport. Foundations and Trends® in Machine Learning. 2019; 11(5-6):355–607. https://doi.org/10.1561/2200000073

65. Wu L, Yen IEH, Xu K, Zhao L, Xia Y, Witbrock M. From Node Embedding to Graph Embedding: Scalable Global Graph Kernel via Random Features; 2018.

66. Xu H, Luo D, Zha H, Carin L. Gromov-wasserstein learning for graph matching and node embedding. arXiv preprint arXiv:190106003. 2019.

67. Maretic HP, Gheche ME, Chierchia G, Frossard P. GOT: An Optimal Transport framework for Graph comparison; 2019.

68. Titouan V, Courty N, Tavenard R, Laetitia C, Flamary R. Optimal Transport for structured data with application on graphs. In: International Conference on Machine Learning; 2019. p. 6275–6284.

69. Bhatia R, Gaubert S, Jain T. Matrix versions of the Hellinger distance. Letters in Mathematical Physics. 2019; p. 1–28.

70. Bhatia R, Jain T, Lim Y. On the Bures–Wasserstein distance between positive definite matrices. Expositiones Mathematicae. 2019; 37(2):165–191. https://doi.org/10.1016/j.exmath.2018.01.002

71. Rezaeinia P, Fairley K, Pal P, Meyer FG, Carter RM. Identifying Brain Network Topology Changes in Task Processes and Psychiatric Disorders. Network Neuroscience. 2020; 0(ja):1–27.

72. Papadimitriou CH. Computational Complexity. In: Encyclopedia of Computer Science. John Wiley and Sons Ltd.; 2003. p. 260–265. Available from: http://dl.acm.org/citation.cfm?id=1074100.1074233.

73. Halko N, Martinsson P, Tropp J. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. SIAM Review. 2011; 53(2):217–288. https://doi.org/10.1137/090771806

74. Bondy JA, Murty USR. Graph Theory. Berlin, Heidelberg: Springer-Verlag; 2008.

75. Wills P. The NetComp Python library; 2019. https://www.github.com/peterewills/netcomp.

76. Ballobás B. Random Graphs. Cambridge University Press; 2001.

77. Zhou D, Huang J, Schölkopf B. Learning from Labeled and Unlabeled Data on a Directed Graph. In: Proceedings of the 22nd International Conference on Machine Learning. New York, NY, USA: ACM; 2005. p. 1036–1043.

78. Hagberg AA, Schult DA, Swart PJ. Exploring network structure, dynamics, and function using NetworkX. In: Proceedings of the 7th Python in Science Conference (SciPy2008). Pasadena, CA USA; 2008. p. 11–15.

79. Abbe E, Bandeira AS, Hall G. Exact recovery in the stochastic block model. IEEE Transactions on Information Theory. 2016; 62(1):471–487. https://doi.org/10.1109/TIT.2015.2490670

80. Yule GU. A Mathematical Theory of Evolution, based on the Conclusion of Dr. J. C. Willis, F.R.S. Philisophical Transactions of the Royal Society B. 1925; 213:402–410.

81. Durrett R. Random graph dynamics. vol. 200 (7). Cambridge university press Cambridge; 2007.

82. Van Der Hofstad R. Random graphs and complex networks. vol. 1. Cambridge university press; 2016.

83. Flaxman A, Frieze A, Fenner T. High Degree Vertices and Eigenvalues in the Preferential Attachment Graph. In: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques: 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2003 and 7th International Workshop on Randomization and Approximation Techniques in Computer Science, RANDOM 2003, Princeton, NJ, USA, August 24-26, 2003. Proceedings. Springer Berlin Heidelberg; 2003. p. 264–274. Available from: https://doi.org/10.1007/978-3-540-45198-3_23.

84. Bender EA, Canfield ER. The asymptotic number of labeled graphs with given degree sequences. Journal of Combinatorial Theory, Series A. 1978; p. 296–307. https://doi.org/10.1016/0097-3165(78)90059-6

85. Zhang X, Nadakuditi RR, Newman MEJ. Spectra of random graphs with community structure and arbitrary degrees. Phys Rev E. 2014; 89:042816. https://doi.org/10.1103/PhysRevE.89.042816

86. Bayati M, Kim JH, Saberi A. A Sequential Algorithm for Generating Random Graphs. Algorithmica. 2010; 58(4):860–910. https://doi.org/10.1007/s00453-009-9340-1

87. Blitzstein J, Diaconis P. A sequential importance sampling algorithm for generating random graphs with prescribed degrees. Internet mathematics. 2011; 6(4):489–522. https://doi.org/10.1080/15427951.2010.557277

88. Exponential Random Graph Models for Social Networks: Theory, Methods, and Applications. Cambridge University Press; 2012.

89. Wolfe PJ, Olhede SC. Nonparametric graphon estimation. arXiv preprint arXiv:13095936. 2013.

90. Li Y, Vinyals O, Dyer C, Pascanu R, Battaglia P. Learning deep generative models of graphs. arXiv preprint arXiv:180303324. 2018.

91. You J, Ying R, Ren X, Hamilton WL, Leskovec J. Graphrnn: Generating realistic graphs with deep auto-regressive models. arXiv preprint arXiv:180208773. 2018.

92. Shine A, Kempe D. Generative Graph Models based on Laplacian Spectra? In: The World Wide Web Conference. ACM; 2019. p. 1691–1701.

93. Dehmamy N, Barabási AL, Yu R. Understanding the Representation Power of Graph Neural Networks in Learning Graph Topology. arXiv preprint arXiv:190705008. 2019.

94. Stehlé J, Voirin N, Barrat A, Cattuto C, Isella L, Pinton JF, et al. High-resolution measurements of face-to-face contact patterns in a primary school. PloS one. 2011; 6(8):e23176. https://doi.org/10.1371/journal.pone.0023176 PMID: 21858018

95. Leskovec J, Kleinberg J, Faloutsos C. Graph evolution: Densification and shrinking diameters. ACM Transactions on Knowledge Discovery from Data (TKDD). 2007; 1(1):2. https://doi.org/10.1145/1217299.1217301

96. Di Martino A, Yan CG, Li Q, Denio E, Castellanos FX, Alaerts K, et al. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. Molecular Psychiatry. 2013; 19:659–667. https://doi.org/10.1038/mp.2013.78 PMID: 23774715

97. Leskovec J. Stanford Large Network Dataset Collection; 2014. http://snap.stanford.edu/data/.

98. Sporns O, Chialvo DR, Kaiser M, Hilgetag CC. Organization, development and function of complex brain networks. Trends in cognitive sciences. 2004; 8(9):418–425. https://doi.org/10.1016/j.tics.2004.07.008 PMID: 15350243

99. Sporns O, Betzel RF. Modular brain networks. Annual review of psychology. 2016; 67:613–640. https://doi.org/10.1146/annurev-psych-122414-033634 PMID: 26393868

100. Fornito A, Zalesky A, Bullmore E. Fundamentals of Brain Network Analysis. Academic Press; 2016.

101. Subbaraju V, Suresh MB, Sundaram S, Narasimhan S. Identifying differences in brain activities and an accurate detection of autism spectrum disorder using resting state functional-magnetic resonance imaging: A spatial filtering approach. Medical Image Analysis. 2017; 35:375–389. https://doi.org/10.1016/j.media.2016.08.003 PMID: 27585835

102. Fornito A, Zalesky A, Pantelis C, Bullmore ET. Schizophrenia, neuroimaging and connectomics. NeuroImage. 2012; 62(4):2296–2314. https://doi.org/10.1016/j.neuroimage.2011.12.090. PMID: 22387165

103. Autism Brain Imaging Data Exchange;. http://fcon_1000.projects.nitrc.org/indi/abide/.

104. Casella G, Berger RL. Statistical inference. vol. 2. Duxbury Pacific Grove, CA; 2002.

105. Hajij M, Wang B, Scheidegger C, Rosen P. Visual detection of structural changes in time-varying graphs using persistent homology. In: 2018 IEEE Pacific Visualization Symposium (PacificVis). IEEE; 2018. p. 125–134.

106. European Parliament Plenary Debates; 2003. https://www.europarl.europa.eu/sides/getDoc.do?type=CRE&reference=20031008&secondRef=TOC&language=EN.

107. Gutfraind A, Safro I, Meyers LA. Multiscale network generation. In: 2015 18th International Conference on Information Fusion (Fusion). IEEE; 2015. p. 158–165.

108. Von Luxburg U. A tutorial on spectral clustering. Statistics and computing. 2007; 17(4):395–416. https://doi.org/10.1007/s11222-007-9033-z

109. Burgess GC, Kandala S, Nolan D, Laumann TO, Power JD, Adeyemo B, et al. Evaluation of Denoising Strategies to Address Motion-Correlated Artifacts in Resting-State Functional Magnetic Resonance Imaging Data from the Human Connectome Project. Brain Connectivity. 2016; 6(9):669–680. https://doi.org/10.1089/brain.2016.0435 PMID: 27571276

110. Primary school temporal network data;. http://www.sociopatterns.org/datasets/primary-school-temporal-network-data/.

111. Meyer FG, Shen X. Perturbation of the eigenvectors of the graph Laplacian: Application to image denoising. Applied and Computational Harmonic Analysis. 2014; 36(2):326–334. https://doi.org/10.1016/j.acha.2013.06.004

112. Cheung G, Magli E, Tanaka Y, Ng MK. Graph spectral image processing. Proceedings of the IEEE. 2018; 106(5):907–930. https://doi.org/10.1109/JPROC.2018.2799702

113. Saggar 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. https://doi.org/10.1038/s41467-018-03664-4 PMID: 29643350

114. Geniesse C, Sporns O, Petri G, Saggar M. Generating dynamical neuroimaging spatiotemporal representations (DyNeuSR) using topological data analysis. Network Neuroscience. 2019; 3(3):763–778. https://doi.org/10.1162/netn_a_00093 PMID: 31410378

115. Chung MK, Lee H, Ombao H, Solo V. Exact topological inference of the resting-state brain networks in twins. Network Neuroscience. 2019; 0(0):1–22.

116. Tzourio-Mazoyer N, Landeau B, Papathanassiou D, Crivello F, Etard O, Delcroix N, et al. Automated Anatomical Labeling of Activations in SPM Using a Macroscopic Anatomical Parcellation of the MNI MRI Single-Subject Brain. NeuroImage. 2002; 15(1):273–289. https://doi.org/10.1006/nimg.2001.0978. PMID: 11771995

117. Rasero J, Aerts H, Ortega MO, Cortes JM, Stramaglia S, Marinazzo D. Predicting functional networks from region connectivity profiles in task-based versus resting-state fMRI data. PloS one. 2018; 13(11):e0207385. https://doi.org/10.1371/journal.pone.0207385 PMID: 30419063

118. Redcay E, Moran J, Mavros P, Tager-Flusberg H, Gabrieli J, Whitfield-Gabrieli S. Intrinsic functional network organization in high-functioning adolescents with autism spectrum disorder. Frontiers in Human Neuroscience. 2013; 7:573. https://doi.org/10.3389/fnhum.2013.00573 PMID: 24062673

119. Hull JV, Jacokes ZJ, Torgerson CM, Irimia A, Van Horn JD. Resting-State Functional Connectivity in Autism Spectrum Disorders: A Review. Frontiers in Psychiatry. 2016; 7:205. https://doi.org/10.3389/fpsyt.2016.00205 PMID: 28101064

120. Roli F, Giacinto G, Vernazza G. Methods for Designing Multiple Classifier Systems. In: Multiple Classifier Systems: Second International Workshop, MCS 2001 Cambridge, UK, July 2–4, 2001 Proceedings. Springer Berlin Heidelberg; 2001. p. 78–87. Available from: https://doi.org/10.1007/3-540-48219-9_8.