Network Science **9** (3): 328–353, 2021
doi:10.1017/nws.2021.11

CAMBRIDGE UNIVERSITY PRESS

ORIGINAL ARTICLE

Diffusion profile embedding as a basis for graph vertex similarity

Scott Payne[1], Edgar Fuller[2*], George Spirou[3] and Cun-Quan Zhang[1]

[1]Department of Mathematics, West Virginia University, Morgantown, WV, USA (e-mails: spayne7@mix.wvu.edu, cun-quan.zhang@mail.wvu.edu), [2]Department of Mathematics and Statistics, Florida International University, Miami, FL, USA and [3]Department of Medical Engineering, University of South Florida, Tampa, FL, USA (e-mail: gspirou@usf.edu)
*Corresponding author. Email: ejfuller@gmail.com

Action Editor: Ulrik Brandes

Abstract

We describe here a notion of *diffusion similarity*, a method for defining similarity between vertices in a given graph using the properties of random walks on the graph to model the relationships between vertices. Using the approach of graph vertex embedding, we characterize a vertex v_i by considering two types of diffusion patterns: the ways in which random walks emanate from the vertex v_i to the remaining graph and how they converge to the vertex v_i from the graph. We define the similarity of two vertices v_i and v_j as the average of the cosine similarity of the vectors characterizing v_i and v_j. We obtain these vectors by modifying the solution to a differential equation describing a type of continuous time random walk.

This method can be applied to any dataset that can be assigned a graph structure that is weighted or unweighted, directed or undirected. It can be used to represent similarity of vertices within community structures of a network while at the same time representing similarity of vertices within layered substructures (e.g., bipartite subgraphs) of the network. To validate the performance of our method, we apply it to synthetic data as well as the neural connectome of the *C. elegans* worm and a connectome of neurons in the mouse retina. A tool developed to characterize the accuracy of the similarity values in detecting community structures, *the uncertainty index*, is introduced in this paper as a measure of the quality of similarity methods.

Keywords: similarity function; vertex similarity; diffusion profile; random walks; diffusion kernel

1. Introduction

The notion that patterns of diffusion across a network reflect structural features embedded in its connectivity is ubiquitous in the field of computational network science. Katz's status index based on random walks was the first sophisticated way to measure the relative importance of a given network vertex (Katz, 1953), now often referred to as centrality. The same form of random walk used by Katz is the basis of the PageRank algorithm, arguably the most important diffusion method in network theory as it led to the organization of the modern internet (Page et al., 1999). More recently, Markov stability (Delvenne et al., 2013) has been shown to generalize the concept of community structure using the induced bottleneck effect of diffusion, and the InfoMap method (Rosvall & Bergstrom, 2008) based on the map equation (Rosvall et al., 2009) has been shown to capture community structure encoded by random walkers' pathways. These are only a few examples, but they illustrate the diverse ways that diffusion has shaped our understanding of what it means to have organized structure in a network. A useful review of many other examples of diffusion in networks is found in (Masuda et al., 2017).

The work presented in this paper approaches the question of network organization and its relation to diffusion through the concept of similarity of network vertices. The notion of similarity is prevalent in data science (Frey & Dueck, 2007; Shirkhorshidi et al., 2015). Clustering and classification, for example, are fundamental tasks that rely on similarity or dissimilarity between individual data points to identify group structure in data. For the application of such methods in the context of network structures, methods to quantify vertex similarity are extremely useful. We might expect a network with organized structure to have groups of vertices which are more similar within a group but less similar between groups. Many methods have been explored to address the question of vertex similarity, some directly using methods such as kernels (Jaccard, 1901; Salton & McGill, 1983; Leicht et al., 2006; Kondor & Lafferty, 2002; Smola & Kondor, 2003; Cooper & Barahona, 2010; Fouss et al., 2006), and some indirectly via graph distances or embedding techniques such as graph Laplacian embedding (Lenart, 1998; Fiedler, 1989; Chan et al., 1994; Shi & Malik, 2000; Meila & Shi, 2001; Perrault-joncas & Meila, 2011; Luo et al., 2009; Fouss et al., 2007; Bai et al., 2005; Ghawalby & Hancock, 2015; Huang et al., 2015; Cheng et al., 2019), or inference approaches such as link prediction (Liben-Nowell & Kleinberg, 2007; Zhou et al., 2009; Lü & Zhou, 2011; Pech et al., 2019; Zhou et al., 2021). Indeed, the position of vertices in networks can be generalized into approaches that identify deeper mathematical properties related to vertex configurations as in (Brandes, 2016), which provides a general framework for the theoretical idea of vertex similarity. Importantly, many proposed definitions have relied on different interpretations of diffusion (Leicht et al., 2006; Kondor & Lafferty, 2002; Smola & Kondor, 2003; Bai et al., 2005; Cooper & Barahona, 2010; Estrada & Silver, 2017; Fouss et al., 2006; Meila & Shi, 2001; Cheng et al., 2019; Thiel & Berthold, 2010; Fouss et al., 2007; Ghawalby & Hancock, 2015; Huang et al., 2015). All of these techniques have important applications ranging from the identification of important elements of social networks (Lü et al., 2011; Qi et al., 2013) to providing insight into complex biological structures (Kovács et al., 2019; Qi et al., 2011).

An early work that provides flexibility for generalizing the notion of diffusion similarity is the diffusion kernel (Kondor & Lafferty, 2002; Smola & Kondor, 2003). A review of several types of diffusion-related kernel similarities and non-kernel similarities for network vertices is found in (Avrachenkov et al., 2019). Kernels are a widely used technique for quantifying similarity between data points in general, since a kernel function generates an implicit embedding of the data in a Hilbert space where similarity is the inner product of the embedded vectors and thus has convenient mathematical properties (Genton, 2002; Shawe-Taylor & Cristianini, 2004).

It is important to note, however, that when employing a kernel method, the inner product property is not by itself enough to ensure that a given kernel similarity is useful since the implied embedding is not automatically guaranteed to relate to the features of the data we might wish to represent. In many diffusion-related similarities for network vertices, the heuristic explanation providing the relevance of the kernel embedding is that the i, j entry of the similarity matrix is related to a scaled sum over i, j random walks of various lengths (Avrachenkov et al., 2019). In particular, the heat kernel similarity (Kondor & Lafferty, 2002) yields a matrix where the i, j entry is the amount of heat at vertex i after time t given that one unit of heat was initialized at vertex j. Thus, in the case of the heat kernel, there is a useful physical interpretation for the way that this similarity captures the organizational properties of a given network: vertices closer in walking distance would share more heat than those further away and, importantly, we would expect dense communities separated by sparse connectivity to trap heat. This bottlenecking effect when heat diffusion occurs in a network with community structure is mathematically expressed, at the global level, by the eigenmode corresponding to the second smallest eigenvalue of the graph Laplacian matrix (Ghawalby & Hancock, 2015; Fiedler, 1989). For recent results on this property and a discussion of related work, see (Cheng et al., 2019).

In this work, we define a specific diffusion profile similarity and develop a framework wherein a given similarity measure can be studied in terms of its ability to represent various models of organized structure. Using our experimental framework, we assess the behavior of our diffusion profile

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

similarity as it relates to changes in basic structured network model parameters. Specifically, our similarity method assigns each vertex to the diffusion pattern it generates in a model of diffusion that has been explored in the literature as *heat kernel pagerank* (Chung, 2007) of a vertex in a graph and also through *unbiased* and *continuous time random walks* (Delvenne et al., 2013; Masuda et al., 2017; Petit et al., 2019; Angstmann et al., 2013) beginning at a vertex. Beginning from several of the principles in these prior works, we adapt the solution to the relevant ordinary differential equation to model diffusion on a network in a way that detects both the properties of the neighborhood of vertices in a community and layered or parallel structures in a network within the diffusion patterns into and out of the vertex. We then compute the similarity of two vertices as the normalized inner product (vector cosine) of their diffusion patterns. The strategy of defining vertex similarity in terms of dispersion patterns represented by diffusion vectors has precedent, for example, see (Thiel & Berthold, 2010; Cooper & Barahona, 2010; Huang et al., 2015). While the design of our similarity relies on a combination of methods that have been applied in a number of other analytic works, we find through experimentation that the specific details of the approach presented here such as the choice of diffusion model, fixed choice of time parameter, and other modifications make a significant difference in terms of performance with respect to structure models. To demonstrate the efficacy of this method, we apply it to two synthetic datasets as well as to sequences of stochastic block model graphs (SBMs) with parameterized characteristics. We then apply it to the neural connectome of the *C. elegans* nematode and use it to identify structures within this biological network. Finally, we apply our method to the mouse retina connectome dataset e2006 and identify modules of relative high similarity that correspond to functional biological modules there.

2. Notation and terminology

We follow the graph theory notation and terminology from standard textbooks, such as, (Bondy & Murty, 2008), (Diestel, 2017), and (West, 2001). An undirected or directed graph G consists of a pair of sets, the vertex set V, and the edge set or directed edge (arc) set $E(G)$ (sometimes we may write simply E). It is denoted by $G = (V, E)$. For a vertex v of an undirected graph, a neighbor u of v is a vertex adjacent to v. The set of all neighbors of v is denoted by $N(v)$. For a vertex v of a directed graph, an out-neighbor (or in-neighbor) u of v is a vertex dominated by v (dominating v). The set of all out-neighbors (in-neighbors) of v is denoted by $N^+(v)$ (or, $N^-(v)$, respectively).

For a weighted, undirected graph G with n vertices, the adjacency matrix A is an $(n \times n)$-matrix where the (i, j) entry $A[i, j]$ is the weight of the edge between v_i and v_j (if G is unweighted, $A[i, j] = 1$ if $v_i v_j \in E(G)$ and $= 0$ otherwise). If the graph is directed then $A[i, j] \neq 0$ means, the arc $v_i v_j$ is in the direction from v_i to v_j.

A walk from a vertex x to a vertex y in a graph G is a collection of edges $\{e_i\}$ from the vertex x to the vertex y identified by a vertex sequence $v_0 v_1 v_2 \cdots v_\ell$ where $v_i v_{i+1} = e_i \in E(G)$. ℓ is the length of the walk.

Let M be an $(n \times n)$-matrix. The (i, j)-term of M is denoted by $M[i, j]$. The i-th row (or i-th column) of the matrix M is denoted by $\vec{M}_{\text{row}}(i)$ (or $\vec{M}_{\text{col}}(i)$, respectively).

3. Diffusion profile similarity method

3.1 Algorithm

Here we state precisely our chosen method of calculating diffusion profile similarity. Details and mathematical relevance of each step will be further discussed in the next subsections. The algorithm proceeds as shown in Figure 1.

Input. The algorithm takes as its input an adjacency matrix A of an input graph G where the (i, j)-entry $A[i, j]$ is the weight of the (possibly) directed edge $v_i \to v_j$ and $n = |V(G)|$. (If G is undirected, A is symmetric; if G is unweighted, $A[i, j] = 0$ or 1.)

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

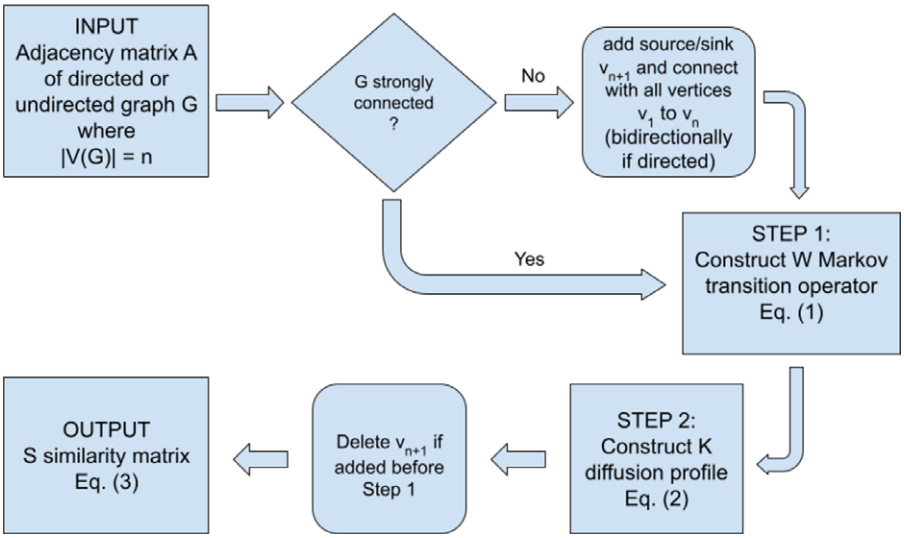

Figure 1. Flowchart of Algorithm from Section 3.1: from adjacency matrix *A* to similarity matrix *S*.

Output. The result is a diffusion similarity matrix S where the (i, j)-entry $S[i, j]$ is the similarity between the vertices v_i and v_j.

NOTE: The completion of **Step 1** is in general recommended for any directed graph input in order to avoid the additional computation time of directly checking the graph to see if it is strongly connected.

Step 1. If G is strongly connected, then go to Step 2. Otherwise, let v_{n+1} be a new vertex, and

$$V(G) \leftarrow V(G) \cup \{v_{n+1}\},$$

$$E(G) \leftarrow E(G) \cup \{v_i \rightarrow v_{n+1} : i = 1, \cdots, n\} \cup \{v_{n+1} \rightarrow v_i : i = 1, \cdots, n\}$$

and for every $i = 1, \cdots, n$,

$$A[i, n+1] = A[n+1, i] \; = \; 10\% \min\{A[i, j] : A[i, j] > 0, \forall i, j = 1, \cdots, n\}$$

then go to Step 2.

Step 2. Construct W where the (i, j)-entry

$$W[i, j] := \frac{A[i, j]}{\sum_{\forall \mu} A[i, \mu]} \tag{1}$$

Step 3. Construct

$$K := \sum_{k=1}^{\infty} \frac{(\approx 1.63)^k}{k!} W^k \tag{2}$$

Step 4. If $|V(G)| = n + 1$ (that is, Step 1 was taken), then deleting the $(n+1)$st-row and $(n+1)$st-column from K. And go to Step 5.

If $|V(G)| = n$, then go to Step 5.

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

Step 5. Let $\vec{K}_{row}(i)$ be the i-th-row of K and $\vec{K}_{col}(i)$ be the i-th-column of K. Construct the output S where the (i, j)-entry

$$S[i,j] := \frac{\cos\left(\vec{K}_{col}(i), \vec{K}_{col}(j)\right) + \cos\left(\vec{K}_{row}(i), \vec{K}_{row}(j)\right)}{2} \tag{3}$$

where $\cos(\vec{\alpha}, \vec{\beta})$ is the cosine of the angle between vectors $\vec{\alpha}$ and $\vec{\beta}$. **END.**

3.2 Relation of this method to notions of diffusion on networks

3.2.1 Relation to Markov Processes

The transition probability matrix (Markov transition operator) W (defined in Equation (1)) is the basis of diffusion in the PageRank algorithm (Page et al., 1999) as well as Markov Stability optimization (Delvenne et al., 2013) and the "heat kernel pagerank" method (Chung, 2007). In its general form $W[i,j] = \frac{A[i,j]}{\sum_{\forall \mu} A[i,\mu]}$, it induces a step in a simple random walk (discrete time diffusion) on a weighted/unweighted, directed/undirected network via the operation:

$$p_{t+1} = W^T p_t \tag{4}$$

where $p_t \in \mathbb{R}^{|V|}$ ($n \times 1$ vector) is the probability distribution on the vertices at step t (Lovász, 1993). Hence, $W[i,j]$ is the probability of reaching v_j directly from v_i (i.e., in one step). W^k computes the random walk distribution at step k and so $W^k[i,j]$ is equal to the following probability,

$$P(\text{ the random walk of length } k \text{ ended at } v_j \mid \text{the random walk of length } k \text{ started at } v_i).$$

The i, j entry of K Equation (2) then is a scaled sum of arrival probabilities from i to j at various step lengths that favors shorter walks. Edge weights in a weighted network augment the transition probabilities at each step. Rows and columns of K, Equation (2), are encodings (embeddings) representing the way v_i and v_j (respectively) relate to the rest of the network via random walk interaction. We can expect that such encodings would reflect models of organized structure in a network, for example, in a network with dense block (community) structure it is understood that there will be similarity between the encodings (embeddings) of v_i and v_j when they are members of the same community as long as the edge set connecting the community to the rest of the network is sufficiently sparse.

The technique of encoding relationships between vertices by a scaled sum over random walks of all lengths is a heuristic that dates back to the earliest work in this field (Katz, 1953) as well as more recent proposals for vertex similarity (Leicht et al., 2006). In this work, we add to this approach by recognizing that the columns and rows of matrix K Equation (2) are diffusion patterns that can be understood through the relationship of K to the Markov transition operator of a random walk process known as active, node-centric continuous time random walk (Petit et al., 2019; Masuda et al., 2017). In this process, a random walker at vertex i takes a step to vertex j with probability $\frac{A[i,j]}{\sum_{\forall \mu} A[i,\mu]}$ and that event occurs after a waiting time t which is an exponentially distributed random variable. Equation (5) is known as the *master equation* of this continuous time random walk (Petit et al., 2019; Angstmann et al., 2013) where the $(n \times 1)$ function of time y is the probability distribution on vertex set V, sometimes referred to as the *residence probabilities* (Petit et al., 2019):

$$\frac{\mathrm{d}}{\mathrm{d}t} y = -(I - W^{\mathrm{T}}) y \tag{5}$$

The matrix $\Phi(t)$ of Equation (6) is the fundamental solution (Brauer & Nohel, 1969) to the first-order linear system in Equation (5) and computes the residence probability distribution y at time t via the operation $y = \Phi(t) y_0$ where y_0 is the distribution at $t = 0$, hence $\Phi(t)$ is a continuous time

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

Markov transition operator:

$$\Phi(t) = e^{-(I-W^T)t} = e^{-t}\sum_{k=0}^{\infty}\frac{t^k}{k!}(W^T)^k \tag{6}$$

Column j of $\Phi(t)$ is the distribution at time t given that the walk began at vertex j (at $t=0$) and row i of $\Phi(t)$ are the residence probabilities of vertex i at time t for each of the possible starting vertices.

As the coefficient e^{-t} is a scalar, Equation (6) is equivalent to K, Equation (2), in the approach presented here with two exceptions. First, the matrix is transposed but this has no effect on our method due to Step 5, Equation (3). Second, and perhaps most importantly, the start term of the series in Equation (6) is $k=0$, while the series in Equation (2) begins at $k=1$. This choice of start term turns out to be advantageous and is discussed below. By the above observations, our diffusion profile similarity method is a similarity of the dispersion profiles of continuous time random walk except for a "reduced" diagonal on account of our start term $k=1$.

3.2.2 Continuous Time Random Walk vs Heat Diffusion

The phrase "heat kernel"(Kondor & Lafferty, 2002; Smola & Kondor, 2003) is derived from the classical study of heat flows. An early use of the dynamical system Equation (5) is found in (Chung, 2007) in which the solution matrix (6) is used in much the same way as the PageRank (Page et al., 1999) matrix to find local clusters around a vertex in a graph. In that work (Chung, 2007), the local Cheeger constant (Mohar, 1989) is used to prove that information about local clustering is encoded in the columns of 6. As a part of that development, Equation (6) is referred to as "heat kernel pagerank" since the base matrix $I-W$ of PageRank (Page et al., 1999) is instead employed in a matrix exponential, a technique commonly referred to as a kernel.

It is important to note two things however. First, Equation (6) is not a true kernel matrix (positive semi-definite matrix) (Avrachenkov et al., 2019) as $I-W$ is not symmetric unless the graph G is degree regular. Also Equation (6) is not generally symmetric. Second, the diffusion defined by Equation (5) does not have the "temperature/electrical model property" described in (Kondor & Lafferty, 2002). It is, however, easy to see that given diagonal degree matrix D we have $D^{1/2}(I-W)D^{-1/2}=\mathscr{L}$, where $\mathscr{L}$ is the normalized graph Laplacian (Chung, 1997), and so $(I-W)$ and $\mathscr{L}$ are similar matrices and share the same eigenvaules. If w is an eigenvector of $\mathscr{L}$, then $u=D^{-1/2}w$ is an eigenvector of $(I-W)$. While the resulting set of eigenvectors of $(I-W)$ are not othonormal, they are linearly independent. Hence, there is an easily stated relationship between the solution matrix Equation (6) and the matrix $e^{-\mathscr{L}t}$ which was studied as an eigenvector embedding method in (Bai et al., 2005).

Perhaps most importantly, we prefer the diffusion model defined by Equation (5) for the following reasons:

(1) This diffusion has been proven to represent community structure through bottle neck properties under the frameworks described in (Chung, 2007) and (Delvenne et al., 2013).

(2) We have a heuristic by which to "fix" t as discussed below that performs well, whereas we were not able to achieve such a suitable fixed t or stable performance across various network structure models when we used the "heat kernel" (Kondor & Lafferty, 2002; Smola & Kondor, 2003) diffusion model.

3.3 Some remarks on the properties of the algorithm

In this section, we provide some remarks concerning different aspects of the method for computing S.

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

3.3.1 Motivation for $k = 1$ but not $k = 0$ in Equation (2)

As discussed in Section 3.2.1, in our method, K Equation (2) is equivalent to the matrix described in Equation (6), except that the $k = 0$ term which gives $I = \frac{(tW)^0}{0!}$ has been removed from the summation. We found that for our use with cosine of vector angles as the similarity measure on columns and rows of K, this "reduced diagonal" diffusion matrix outperformed the similarity obtained if we do not reduce the diagonal.

When the diagonal is reduced, the first neighborhood walk terms $K[i, j]$ and $K[j, i]$ are scaled down in the computations of the inner products $\vec{K}_{col}(i) \cdot \vec{K}_{col}(j)$ and $\vec{K}_{row}(i) \cdot \vec{K}_{row}(j)$. This has the effect of increasing the emphasis on the comparison of diffusion flow from vertex v_i to vertices in $V \setminus \{v_i, v_j\}$ with the diffusion flow from vertex v_j to vertices in $V \setminus \{v_i, v_j\}$. In this way, S captures a better picture of the relationship between vertices v_i and v_j relative to the rest of the graph since a high flow between vertices v_i and v_j will not dominate the sum. This reveals information about their flows to other vertices and how the two patterns relate. We have confirmed experimentally that the choice of $k = 1$ as a start value of the sum improves the ability of the similarity method to reflect bipartite substructures and appears to reduce the influence of between-community edges in the detection of community structures.

3.3.2 The Choice of the Parameter $t \approx 1.63$

For the diffusion matrix K in Equation (2), we propose that we should be able to encode information about community structure models as well as common neighborhood models, which focus on first neighborhoods of vertices, as long as the parameter t is chosen to emphasize walks of lengths 1, 2, 3 properly. Let $\beta_k(t) = \frac{t^k}{k!}$ be the coefficient of the k-th term of Equation (2). The value of $\beta_k(t)$ indicates the emphasizing level of the walks of length k in the sum $\sum_{k=1}^{\infty} \frac{t^k}{k!} W^k$. The following is the argument for this selection:

(1) The walks of length 2 cannot be more emphasized than walks of length 1. Hence, $\frac{t}{1!} = \beta_1(t) \geq \beta_2(t) = \frac{t^2}{2!}$. That is, $t \leq 2$ is an upper bound.

(2) The first decrease $\beta_1(t) - \beta_2(t)$ should not be greater than the second decrease $\beta_2(t) - \beta_3(t)$ for otherwise the second term $\beta_2(t)$ is not sufficiently emphasized. That is,

$$\frac{t}{1!} - \frac{t^2}{2!} = \beta_1(t) - \beta_2(t) \leq \beta_2(t) - \beta_3(t) = \frac{t^2}{2!} - \frac{t^3}{3!}$$

and, therefore, t is bounded below by $3 - \sqrt{3}$.

As a result, we choose the average of the upper and lower bounds:

$$t \approx 1.63 \approx \frac{2 + (3 - \sqrt{3})}{2}$$

Experimentally, we have found that t values near to 1.63 work very well in all types of graph data we have tested, whether the graph is large or small, directed or undirected, weighted or unweighted. To provide an even further level of confidence to the heuristic choice of t described here, we also performed "experiment 1" (see Section 4.3.1) repeatedly with t values in the entire interval $(3 - \sqrt{3}, 2)$ and found that the curves relating uncertainty score to between-community edge density were identical. Therefore, given the demonstrated success in the wide range of graph types tested, we are confident that t can safely be treated as a constant instead of a free parameter.

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

3.3.3 S is Positive Semi-definite.

The matrix S of Equation (3) is positive semi-definite since it is the sum of a pair positive-semidefinite matrices, namely $K_c^T K_c$ and $K_r K_r^T$, where K_c is the matrix obtained by multiplying each column of K from Equation (2) by the reciprocal of its norm and K_r is the matrix obtained by multiplying each row of K from Equation (2) by the reciprocal of its norm.

3.3.4 Behavior in the Bipartite Case

Assume that the input graph G contains an induced bipartite subgraph H with high density. One would expect a higher similarity between vertices from the same part of H, and lower similarity for vertices from distinct parts of H. Consider a special case that v_i, v_j have the same set of out-neighbors and the same set of in-neighbors. Then, since v_i and v_j are isomorphic, we have $\vec{K}_{row}(i) \approx \vec{K}_{row}(j)$ and $\vec{K}_{col}(i) \approx \vec{K}_{col}(j)$ (we write "$\approx$" since for each of these pairs of vectors the i-th terms are not equal and j-th terms are not equal, however, equality holds for all other terms). In this way, the similarity S provides a solution to one of the challenges mentioned in the introduction, namely, the representation of various models of organized structures in networks as opposed to representing only dense community structures in these networks.

3.3.5 A Universal Sink/Source

Step 1 of the algorithm is a technical step used to resolve the case where the dynamical system in Equation (5) has degenerate solutions. If a directed graph G has sinks or sources or an undirected graph G has isolated vertices, the matrices W and K will have some all-zero rows or all-zero columns. This will result in zeroes in the denominators in Equation (3) (in the calculation of cosine). A newly added vertex v_{n+1} will avoid the all-zero rows in W and K and therefore resolves this problem. In Step 1 of the algorithm, we state "if G is not strongly connected" because it generalizes both of the degenerate cases described in this section. This technique of adding a universal sink/source has been independently applied by other authors, such as (Lü et al., 2011).

4. Application to synthetic data: Analysis and the uncertainty index

In this section, we consider a number of synthetic graphs and analyze the performance of the similarity matrix S in representing various structures within those networks. First, we present several types of these graphs and then follow this with a developed measure of performance. We then demonstrate the ability of this measure to quantify relationships between accuracy of similarity with respect to planted community structures and parameters of the community model such as edge density.

4.1 Synthetic data case studies

4.1.1 An Example Network with Multiple Subgraph Structures

We present here a small example graph with several key features representing structural properties we wish to detect and then investigate whether diffusion similarity captures these features accurately. This example provides a convenient way to visualize the performance of our similarity measure and to understand the role played by the various components of our calculation. Additionally, we compare the resulting similarity matrix side by side with other options for computing similarities, including cosine similarity and the Leicht–Holme–Newman method (Leicht et al., 2006), in order to verify that our proposed method (the diffusion similarity) does indeed perform at least as well as the other proposed methods examined here.

The synthetic example graph G is illustrated in Figure 2(a) and is the union of two spanning subgraphs G_1 and G_2 as shown in Figure 2(b) and (c). Figure 2(d) is the adjacency matrix of the

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

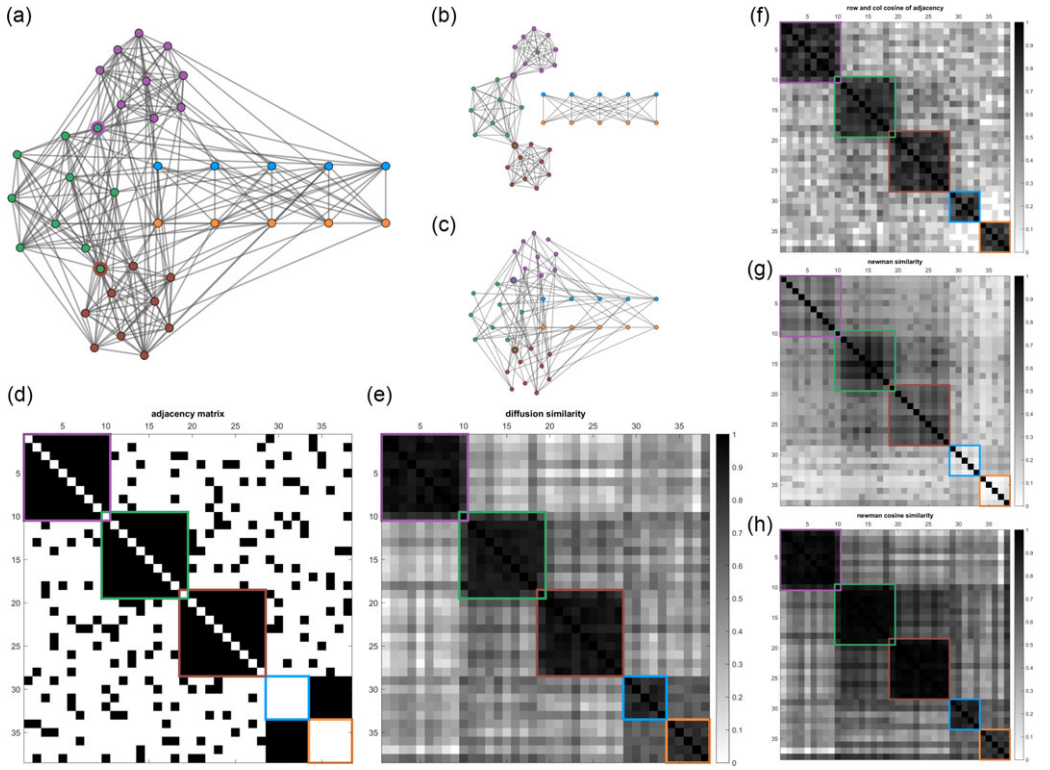

Figure 2. Example: an illustrated graph.

graph G. The spanning subgraph G_1 in Figure 2(b) indicates four parts of the graph G, three cliques (with some small overlaps), and one complete bipartite graph. The edges of another spanning subgraph G_2 illustrated in Figure 2(c) are crossing edges joining these four parts of G_1.

Vertices of Figure 2(a) are color-coded to indicate the designed ground truth output communities of G. The matrix of the diffusion similarity for the graph is seen in Figure 2(e).

1. Functional modules and bipartite subgraphs

In this example, the bipartite subgraph appears in the lower right corner of the adjacency matrix in Figure 2(d). Bipartite structures play an important role in the analysis of networks of neurons, also known as connectomes, as some sets of cells can be described as layered networks. A layer is a set of cells that may have relatively few interconnections but have a common out or in neighborhood of cells forming another layer, and cells in a given layer should be identified as similar. This phenomenon can be seen in various real-world examples such as the functional modules in retinal/visual systems and cochlear/auditory systems. The opposite motif, densely interconnected sets separated by sparse connectivity, also appears in key roles for example as in star amacrine cells in the retina where ON and OFF sets form distinct dense groups. However, most existing similarity methods may not be able to detect bipartite and dense block structures simultaneously. For example, Figure 2(g) is the similarity matrix S_{LHN} produced by Newman method with the bipartite groupings still boxed, but we see that this method shows no community structure for them.

2. The application and effects of the cosine-step

As outlined in Subsection 3.1 and 3.2, our proposed method consists of three steps:

$$A \rightarrow W \rightarrow K \underset{\cos}{\rightarrow} S$$

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

One of the most notable steps is $K \underset{\cos}{\rightarrow} S$ (the *cosine-step*). As discussed in Subsection 3.2.1, $K_{row}(i)$ is a vector characterizing the diffusion out of a give vertex v_i, while $K_{col}(i)$ is a vector characterizing the diffusion arriving at a given vertex v_i. The diffusion similarity $S[i, j]$ between v_i and v_j is defined in Equation (3) to be the average of the cosine similarity of diffusion out-flow distributions and the cosine similarity of diffusion in-flow distributions. The cosine-step plays an important role in the detection of *"bipartite functional modules"*.

If we add the cosine-step to Leicht–Holme–Newman's method (Leicht et al., 2006), that is,

$$A \rightarrow S_{\text{LHN}} \underset{\cos}{\rightarrow} S^*$$

we are able to improve output (see Figure 2(h)) over the usual S_{LHN} (Leicht et al., 2006) (see Figure 2(g)).

Furthermore, if we apply the cosine-step directly to the adjacency matrix A, that is,

$$A \underset{\cos}{\rightarrow} S^{**}$$

we are also able to detect the layers in the planted bipartite structures as indicated by the blue and orange vertex subsets (see Figure 2(f)). Due to the computational costs of power series of matrices or inverse of matrix, this simplified processing S^{**} (although not as good as Figures 2(e) and (h)) is practically useful if one prefers speed to accuracy in certain applications (such as surveillance monitoring). However, as we shall see, these improvements still do not provide better structure detection than S.

3. Multimembership and Hierarchical Structure

The graph G also has two vertices with multi-community membership. As shown in Figure 2 (e), (h), and (f), each of the three presented computations are able to detect such features with similar quality. However, single column/row bands indicate the hierarchical structure of the clustering, and one of the most noticeable clusters is the bipartite subgraph which would appear on the second lowest level of a hierarchical diagram for the graph. In this sense, S outperforms both S^* and S^{**} by delineating the bipartite structure as well as the other multi-community features, while the other two miss significant numbers of vertex to vertex similarity across communities.

4.1.2 A Strictly Directed Example

In this section, we present a directed graph G whose structure is shown in Figure 3 in order to demonstrate that the diffusion-based similarity calculation can reflect organized structures that are independent of models of edge density or sparseness of inter-community connectivity. The adjacency matrix of G is seen in Figure 3(b) and shows the construction of the graph. Within each of the color-labeled groups (blue, red, and green), each pair of vertices is connected by one directed edge going in each direction.

- For any pair of vertices v_i in blue v_j in green, there are exactly two directed edges from v_i blue to v_j green.
- For any pair of vertices v_i in blue v_j in red, there are exactly two directed edges from v_i blue to v_j red.
- For any pair of vertices v_i in green v_j in red, there are exactly two directed edges from v_i green to v_j red.

Thus, each color group is a strongly connected component and for any pair of vertices with one vertex in one group and the other in a different group, the connection goes only in one direction. The specific construction is such that if we ignore edge direction, the graph is precisely an undirected, unweighted complete graph (since a complete graph with all equal edge weights is

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

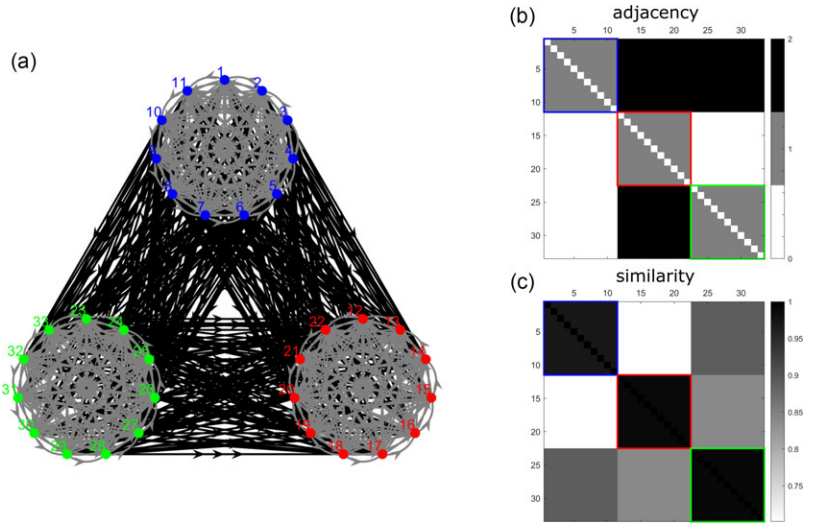

Figure 3. Example: a directed graph with strongly connected components.

equivalent to an unweighted complete graph) and hence has no group structure. The similarity data calculated using Equation (3) are seen in Figure 3(c). This example demonstrates that the diffusion profile similarity is able to represent organized network structure that is embedded strictly in the edge direction with no relationship to edge density. While it is easy to prove that the similarity matrix will be essentially as seen in Figure 3(c) given the isomorphic property of vertices *within a color group of G* (and given the diffusion method described in Section 3.2.1), it is nonetheless useful to see the effectiveness of the similarity computation as shown in this example. When considered along with the example in Section 4.1.1, the two together illustrate how well the diffusion profile similarity method can represent different models of organized structure: directed or undirected, block models or bipartite models. All these structural properties have been revealed by the same calculation in Equation (3).

4.1.3 Similarity within Graphs generated by Stochastic Block Models

As noted above, a key objective for the diffusion similarity measure S (Equation (3)) is to be able to represent community structures within graphs. Graphs generated by SBMs provide a collection of communities of vertices that possess connectivity *inside communities* defined by a likelihood of connection between vertices within those communities and *between-community* connectivity modeled by a usually different probabilistic distribution of edges between these communities. As such, they are increasingly becoming the standard model for community structure and an ideal candidate for synthetic data structures on which to test algorithms for community detection.

Here, we designed an experimental framework to test our similarity measure's performance on these types of synthetic graphs constructed with stochastic blocks. By the strict definition of an SBM graph, each block B_i is constructed by placing an edge between vertices inside the block individually as a Bernoulli trial with probability p_i. Random walk behavior in a graph is controlled by the edges created by that process and not by the likelihood of the existence of the edges, so in order to create sets of graphs with comparable properties we must use a *fixed edge density* model (Frieze & Karoński, 2016) for both the internal and the between-block edge structures. SBM graph blocks are internally an Erdős–Rényi (ER) model (Erdős & Rényi, 1959, 1960), with parameters n_i and p_i, the number of vertices (of the block) and the probability of an edge, respectively. The expected number of edges in a block is $\binom{n_i}{2}p_i$. The edge density of a graph is defined as $\mu = \frac{|E|}{\binom{n}{2}}$,

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

where $|E|$ is the number of edges in the graph. Therefore, a fixed edge ER graph model is the model where a random graph generated with parameters n and p has precisely the number of edges as the expected number in a standard ER graph model.

We may then state the fixed edge SBM model as the model where each block has a number of internal edges given by $\binom{n_i}{2}\mu_i$ and between any pair of blocks B_i and B_j, the number of edges is $n_i n_j \eta_{i,j}$ where $\eta_{i,j}$ is the edge density parameter for edges between blocks B_i and B_j. In order to design experiments that may compare a set of graphs with a varying parameter, we must further refine our model so that for a given graph the parameter $\eta_{i,j}$ is the same value for each pair B_i and B_j. It will be clear in the experimental design why this choice is necessary in order to allow graphs to be comparable (see Subsection 4.3).

Formally, each model graph G that we generate on which we apply our similarity measure has input parameters as follows: block size parameters $n_1, ..., n_k$ where k is the number of blocks, inside-block edge density parameters $\mu_1, ..., \mu_k$ and between-block edge density parameter η.

4.2 The uncertainty index: Measuring the quality of similarity methods

In order to measure how effectively the diffusion similarity function represents the blocks of a graph G generated with a given set of parameter values as described above, we propose to measure the *uncertainty* of the similarity matrix with respect to the set of blocks. In Figure 4, we provide two example graphs to motivate and illustrate the measure of uncertainty. The adjacency matrices of the two graphs, which we refer to as G_a and G_b, are seen in Figures 4(a) and (b), respectively. Graphs G_a and G_b both have 600 vertices, inside-block edge density 75%, and between-block edge density 10%. G_a has three blocks, while G_b has 30 blocks. The diffusion similarity matrix of G_a is seen in Figure 4(c) and the diffusion similarity matrix of G_b is seen in Figure 4(d). Figure 4(e) shows two histograms describing the similarity matrix of G_a with the top indicating the diffusion similarity data measured within blocks and the bottom showing similarity data between blocks. The same histograms for G_b are seen in Figure 4(f). The similarity matrix of G_a represents its blocks more distinctly than G_b in the sense that the inside- and between-block similarity distributions do not overlap. That is, the possible similarity values between two vertices in a block is a different set of values than the possible similarity values for two vertices in different blocks. We can say that the similarity matrix of G_b is *less certain* than that of G_a, since the distributions of inside- and between-block similarities overlap. That is, for a vertex pair v_i, v_j such that $S[i, j]$ is in the overlapping range, the similarity measure does not distinguish whether the pair are in the same block or different blocks. It follows naturally that if there were only a small number of vertex pairs with similarity in the overlapping range, then clustering algorithms relying on similarity might more accurately detect the intended blocks, and in that sense the similarity would be *more certain* with respect to the block structure. To quantify the uncertainty of a similarity measure when there are values in the overlapping range, we measure the overlap of the distributions of inside-block and between-block similarities.

We can estimate the *probability density function* (pdf) of each distribution using the method of kernel density estimation. Let the two pdf curves be the functions ρ_{ins} and ρ_{betw} so that for a given similarity value s, the value $\rho_{\text{ins}}(s)$ is the probability of the value s being a similarity value of two vertices within the same block of the graph G. In the same way, we also have the corresponding definition for $\rho_{\text{betw}}(s)$ for vertices located between blocks. The product of the two functions ρ_{ins} and ρ_{betw} is a curve such that the area under it varies with the overlap of the two distributions. That is, the more likely that similarity values lie within the overlapping range, the higher the *uncertainty* of the similarity matrix with respect to the embedded block structure. Figure 4(g) and (h) show these curves for graphs G_a and G_b, respectively. The red curve is the pdf function ρ_{betw}, the blue curve is the pdf function ρ_{ins}, and the yellow curve is their product. Observe that this uncertainty measure well represents the similarity matrices in Figures 4(c) and (d). The yellow curve in Figure 4(g) shows that the similarity matrix in Figure 4(c) has zero

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

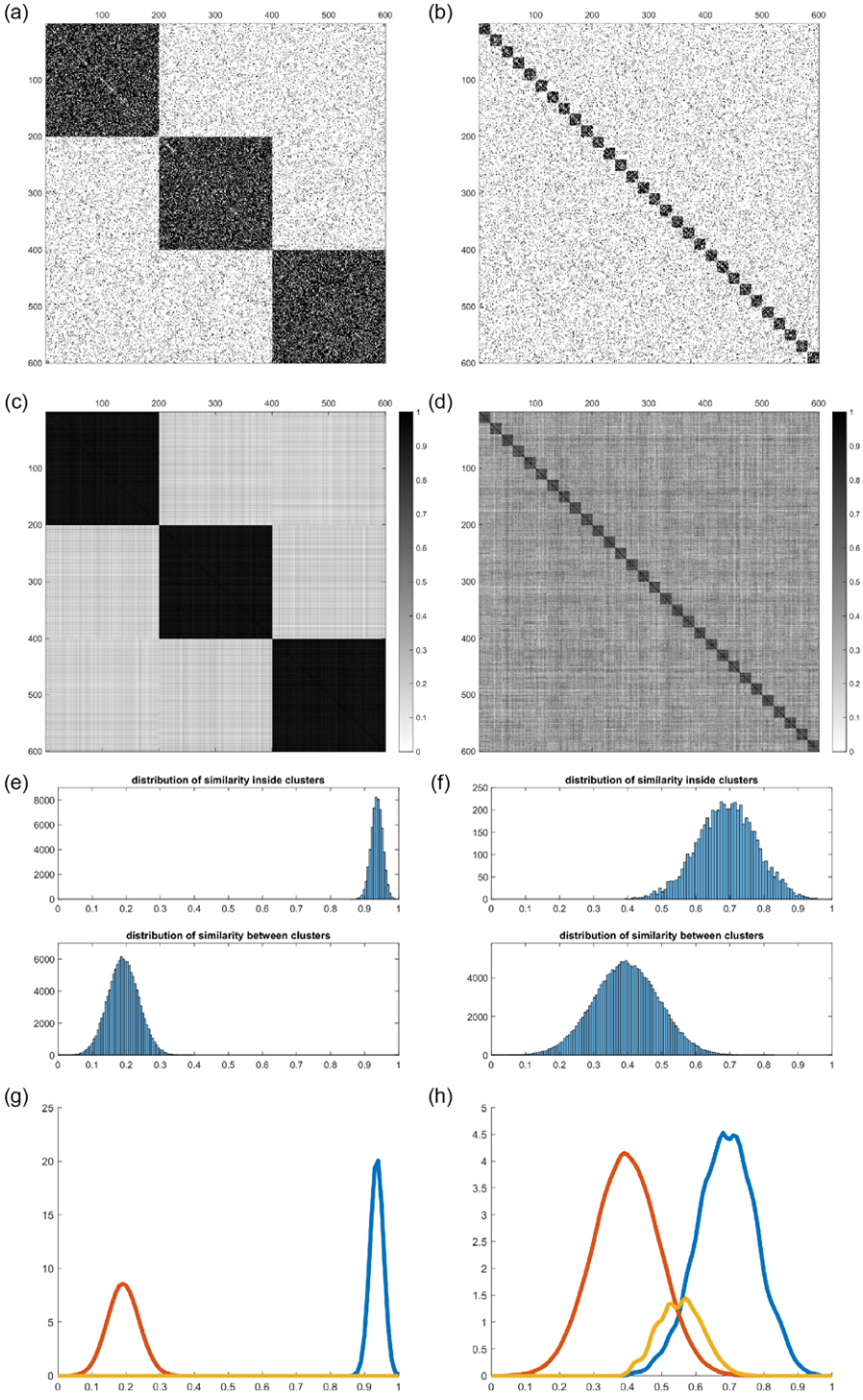

Figure 4. Measuring uncertainty with respect to blocks.

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

uncertainty, while the yellow curve in Figure 4(h) shows that the similarity matrix in d has positive uncertainty. In this way, we define the candidate measure of *uncertainty* is as follows:

$$\int_0^1 \rho_{\text{ins}}(s) * \rho_{\text{betw}}(s)\, \mathrm{d}s \tag{7}$$

The issue that arises with this measure is that it is sensitive to the width of the distributions in a way that affects the maximum value it can achieve. Specifically, when the two distributions are the same (corresponding to maximum uncertainty), the area described by Equation (7) may change with the distribution width. The easy fix for this is to recognize that our measurement is the inner product of the pdfs ρ_{ins} and ρ_{betw}. Since these are stored as discrete arrays computationally, we can measure *uncertainty* as the cosine of the vector angle between the two pdfs which is just their inner product normalized by the product of their norms. This way the maximum uncertainty case will always have value 1 and minimum will be 0, and these will each be achieved when there is 100% and 0% overlap, respectively. We will refer to the following computation as the *uncertainty score* for the similarity matrix S with respect to the blocks:

$$\textit{uncertainty}(S, \text{Blocks}) := \frac{\langle \rho_{\text{ins}}, \rho_{\text{betw}} \rangle}{||\rho_{\text{ins}}|| * ||\rho_{\text{betw}}||} \tag{8}$$

where $\langle \vec{\alpha}, \vec{\beta} \rangle$ is the inner product of the vectors $\vec{\alpha}$ and $\vec{\beta}$.

4.3 Experimental design and results

A key observation about the uncertainty scores of the graphs G_a and G_b (from Section 4.2) informs our experimental design. The graphs have the same number of vertices and same parameters for inside-block edge density and between-block edge density, yet the similarity matrix of G_a has an uncertainty score 0%, while the similarity matrix of G_b has an uncertainty score 7.9%. In this controlled example, we see that the similarity matrix becomes more uncertain as the number of blocks (clusters) goes up for a graph with a fixed number of vertices. Intuitively, this is due to the relative number of connections within a block that are possible compared to the number of between-block connections possible. As the block size decreases but the number of blocks goes up, the ratio of inside-block connections to between-block connections globally decreases, moving the two relevant pdfs closer together. In addition, we note that heuristically uncertainty will increase as the between-block edge density increases in general, since the graph would become more similar to an ER graph where random walks would be fully arbitrary and not be constrained by any block structure. These two principles guide the explorations below.

4.3.1 Experiment 1. The Uncertainty Index with Variable Between-Density

For Experiment 1, we fix the number of vertices at 600 and inside-block edge density at $\approx 74.8\%$ and vary the between-block edge density over a range from 5% to 74%. We perform this test three times, in the first test the graphs generated have 3 blocks, in the second 12 blocks, and in the third 30 blocks. For a given structure type, the block sizes are all the same. For example, in a three-block graph, each block has 200 vertices since the total vertices is 600 for every graph in Experiment 1. We test 200 graphs of each structure type, so the total number of graphs tested in Experiment 1 is 600. Also, note that for each structure type and between-block edge density value, we tested five such graphs in order to study the consistency of the uncertainty measure for a given graph type.

The results are shown in Figure 5(a). Uncertainty increases with between-block edge density and also with the number of blocks. Interestingly, it appears that for each structure type, there is some range where uncertainty grows extremely slowly as between-block edge density increases. For example, graphs with three blocks appear to have uncertainty approximately 0% for

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

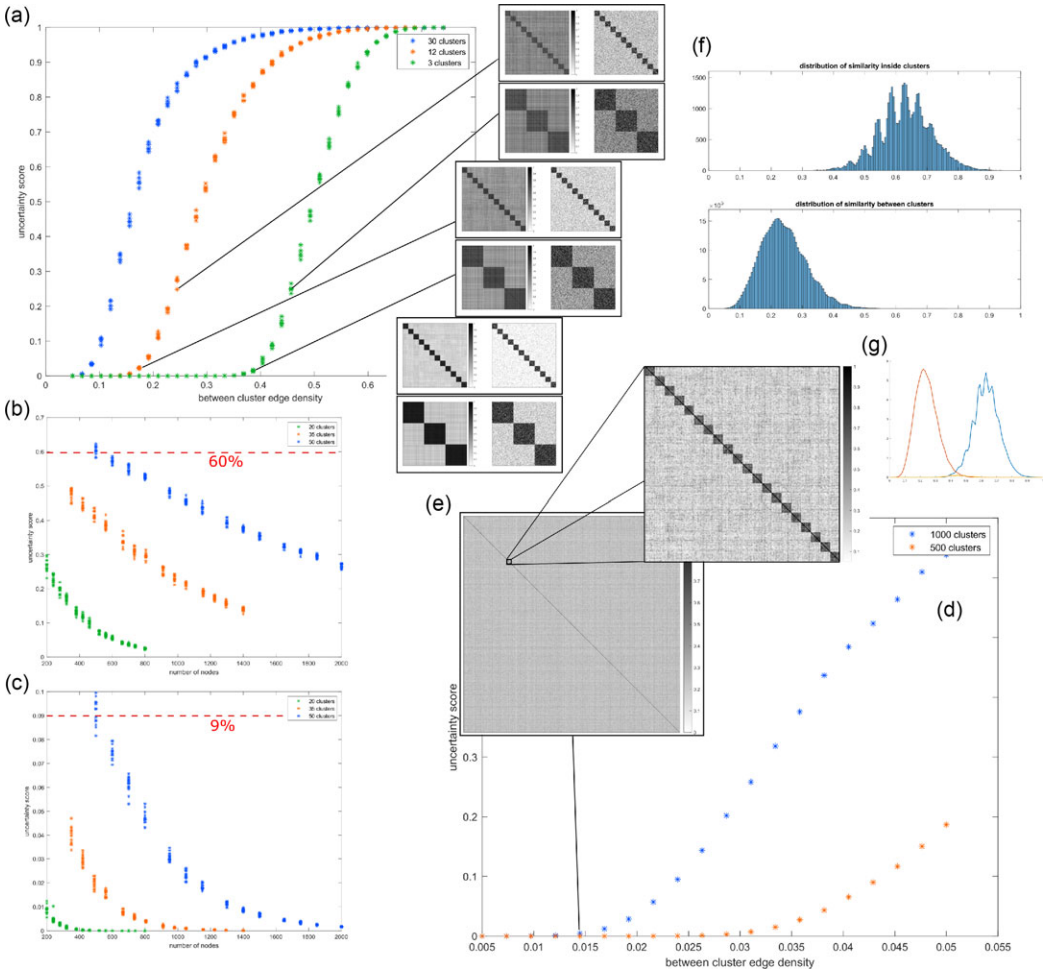

Figure 5. Testing in block model graphs.

all between-block edge density values up to roughly 35%. This would appear to suggest that even when a graph has a large number of blocks, there might be some threshold value of between-block edge density for which the uncertainty of similarity is within some acceptable range.

The diagrams at the right of Figure 5(a) show pairs of graphs having the same uncertainty score. For example, the top square of the four diagrams represents two graphs, one with 12 blocks and one with 3. For each square, the left diagram is the diffusion similarity matrix S (Equation (3)) and the right is the adjacency matrix A. The top two graphs both have uncertainty $\approx 25\%$, the middle two $\approx 2\%$, and the bottom two 0%. Although we cannot establish a proven threshold for which uncertainty would be "acceptable" for data mining tasks in general, visually the diagrams seem to indicate that uncertainty scores higher than 25% might be undesirable. Comparing the six graphs shown, it appears that the uncertainty measure we use performs well in representing the quality of a given similarity matrix with respect to the blocks. We can also use these six graphs to establish an expectation for "good performance" of similarity relative to between-block edge density. Note that one might initially interpret the results of Experiment 1 as showing limited usefulness of similarity, since low uncertainty scores are achieved only for sparse between-block edge density. However, taking the 12 cluster graph in the middle square as an example, we can see that in this graph the average number of edges a vertex has within its block is 36, while the average number

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

of edges a vertex has to vertices outside its block is 95. In fact, the average ratio of between-block degree over inside-block degree is 2.6 and the minimum for that ratio is 1.8 and 98% of vertices have that ratio as at least 2. Since each block has far more external edges than internal edges, we feel that the uncertainty score of 2% actually represents quite good performance of the diffusion similarity measure S (Equation (3)).

4.3.2 Experiments 2 and 3. The Uncertainty Index with a Variable Number of Vertices

Experiments 2 and 3 test the behavior of diffusion similarity relative to the number of vertices in the graph. We fix the number of blocks, inside-block and between-block densities, and generate a sequence of graphs by varying the size of the blocks. Thus, each graph in a sequence has the same structural and density properties, but varying number of vertices. We test three structure types: 20 blocks, 35 blocks, and 50 blocks. For Experiment 2, inside-block edge density is 85% and between-block edge density is 15%. For Experiment 3, inside-block edge density is 85% and between-block edge density is 8%. As in Experiment 1, for each combination of parameter values, we tested several graphs (in this case 14). The results of Experiment 2 are seen in Figure 5(b) and results of Experiment 3 are seen in Figure 5(c). Experiments 2 and 3 demonstrate that increasing graph size actually improves the quality of the similarity measure for a given structure type and edge density parameters. Also, the experiments demonstrate that a dramatic improvement in similarity performance is achieved when between-block edge density is within a favorable range (as in Experiment 3). Interpreting the results of Experiments 1, 2, and 3 together, it appears that the similarity measure might be able to well represent even very large numbers of blocks embedded in very large graphs provided that edge density and block size parameters are "reasonable.'. This conjecture motivates Experiment 4.

4.3.3 Experiment 4. The Uncertainty Index for a Large Number of Blocks

Experiment 4 follows the same general design as Experiment 1, but here using parameters that test the ability to represent very large numbers of blocks within large graphs. We tested a set of 40 graphs each with 10, 000 vertices. In the first subset of 20 graphs, each has 1, 000 blocks, and in the second subset of 20 graphs, each has 500 blocks. Each sequence of 20 graphs represents 20 different between-block edge density values ranging from 0.5% to 5%. For all blocks, the inside-block edge density is 90%. As in Experiment 1, all blocks for a structure type (e.g., 1,000 blocks) have the same size. The results are shown in Figure 5(d). We see that the diffusion similarity measure is able to represent 1, 000 blocks with uncertainty level below 25% as long as between-block edge density is less than about 2.9%.

Figure 5(e) shows the diffusion similarity matrix of a graph with 1, 000 blocks. The between-block edge density is 1.45%. The uncertainty score is 0.43%. The average number of edges a vertex has within its block is 8, while the average number of edges a vertex has to vertices outside its block is 144. The average ratio of between-block degree over inside-block degree is 18.3 and the minimum for that ratio is 11.8. The zoomed inset upper right shows the diffusion similarity of a subgraph consisting of 20 blocks. Figure 5(f) shows the inside-block similarity histogram (top) and between-block similarity histogram (bottom). Figure 5(g) shows the estimated pdf of between-block similarity (red), the estimated pdf of inside-block similarity (blue), and their product function (yellow). We note that the method of measuring uncertainty appears to be working accurately even in the conditions of the large number of values in a $10,000 \times 10,000$-matrix, so the results seem to be consistent across graphs of wide ranging block generation parameters. Importantly, when density parameters are favorable, the similarity measure is able to reflect planted blocks even in graphs with a very large number of blocks that are very small relative to the size of the graph. This ability is a valuable feature for the study of neuron connectomes which can be highly complex and might not be easily described by simple block models.

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

5. Vertex similarity in neural connectomes

Connectomics is the study of the networks of connections found between neurons in the nervous systems of a variety of organisms. Viewing the cell body of a neuron, the soma, as the vertex in a graph and the axon extending from neuron to neuron as an edge in a graph, the synaptic connections between these cells create vastly complex networks. In this section, we apply the similarity matrix construction to two examples. First, we consider the complete connectome of the nematode *Caenorhabditis elegans* (*C. elegans*). We then turn to an analysis of a portion of the retinal connectome of a mouse.

5.1 Application to the central nervous system of C.elegans

C.elegans is a small roundworm, about 1 mm in length, that lives in soil in many regions of the world, and feeds on bacteria (Corsi et al., 2015; Eisenmann, 2005). It was proposed as a model organism by Sydney Brenner (Brenner, 1973), and it has proven useful in many experiments that relate genetics and physiology to animal behavior (Emmons, 2015). *C.elegans* was the first multicellular organism to have its whole genome sequenced and was the first organism to have its entire connectome mapped (White et al., 1986; White, 2013; Jabr, 2012). Recent advances in volume electron microscopy have provided additional connectomes for study (Ohyama et al., 2015; Ryan et al., 2016; Zheng et al., 2018; Ryan et al., 2016; Bock et al., 2011; Briggman et al., 2011; Takemura et al., 2017), increasing the need for tools to identify structure within these complicated networks. In this section, we illustrate our methodology by its application to a subset of the *C.elegans* connectome.

5.1.1 The Dataset and Adjacency Matrix

For this study, we use the complete set of chemical synapse data for this animal as found in (Varshney et al., *n.d.*) (also see (Chen et al., 2006; Chen, 2007; Varshney et al., 2011)). The data are structured as a weighted, directed graph G with 279 vertices (neurons), and total weight 6, 394 in which the weight of each arc (directed edge) from a vertex i to another vertex j is the number of chemical synapses from the neuron i to the neuron j. The induced adjacency matrix A is created in which each $A[i, j]$ is the weight of the arc (directed edge) from i to j.

To simplify this presentation, we apply our methods to subnetworks surrounding two pairs of interneurons, AVBL, AVBR, and AVEL, AVER. These neurons are highly connected to much of the rest of the network and are referred to as command interneurons due to their extensive connections with sensory and motor neurons (Varshney et al., 2011; Towlson et al., 2013; Cook et al., 2019). Lesion studies demonstrate their importance in coordinated motor movement, and from that perspective they are functionally similar (Gray et al., 2005). However, more specifically [AVBL, AVBR] are integral to forward movement and [AVEL, AVER] are integral to backward movement, and these movements can be induced by sensory input to the tail and head regions, respectively (Chalfie et al., 1985). Hence, we might expect the similarity measure also to differentiate between AVB and AVE. Importantly, there is relatively little connectivity within the set [AVBL, AVBR, AVEL, AVER] and so the set represents an instructive example on which to test the ability of our similarity measure to reflect functional layers in a real-world neuron connectome.

These sets of connections are seen in the graphical layout (Figure 6), which employs the method from (Varshney et al., 2011) to place vertices. Blue connections depict the AVBL, AVBR network, and purple connections depict the AVEL, AVER network. Generally, sensory neurons are in the upper half of the diagram and motor neurons are in the lower left. For each connection, direction is indicated as darker color for presynaptic and lighter color for postsynaptic neurons. Hence, the [AVBL, AVBR] and [AVEL, AVER] pairs are primarily presynaptic to neurons at the left. Pictorially, these neuron pairs have generally similar layouts in terms of numbers and locations of inputs and outputs. Neurons within pairs share some inputs and outputs, yet few connections connect [AVB] and [AVE] pairs.

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

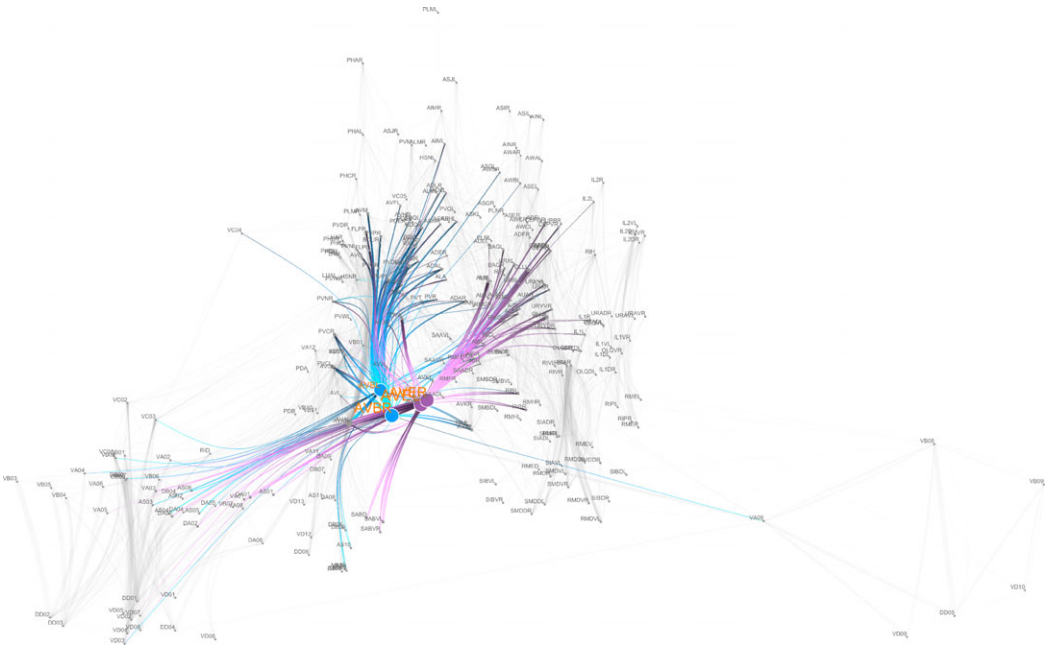

Figure 6. *C.elegans* chemical synapse connectome, displaying in- and out-neighbor connections to AVB pair (blue) and AVE pair (purple). The dark end of each connector is the presynaptic source and the light end of the synaptic contact. So edge direction is from dark to light.

We have ordered the neurons in the induced adjacency matrix A[i, j] to cluster the inputs and outputs of these neuron pairs. The larger blue and purple squares at upper left (Figure 7(c)) enclose the in-neighbors to AVB and AVE pairs, respectively, and the overlap of the squares indicates shared inputs. The smaller blue and purple squares indicate the out-neighbors of the AVB and AVE pairs, colored for each pair, and the overlap indicates common outputs. The small red square indicates neurons that are both in- and out-neighbors of any of AVBL, AVBR and AVEL, AVER. Note that in combination, AVBL, AVBR and AVEL, AVER directly contact a large fraction of neurons in the *C.elegans* nervous system.

5.1.2 Diffusion Similarities Within and Between Pairs

These commonalities and differences in function and network structure should be differentially manifested in the computation of similarity within and across neuron pairs. We illustrate, for each of these four neurons, the diffusion profile vectors used in the similarity calculation. Diffusion profiles for in-flows and out-flows are shown in Figure 7(a) and b, respectively, scaled to the color code at the right of these panels and aligned with the induced adjacency matrix A[i, j]. In the diffusion profile images, vertical alignment of colored cells contributes positively to the similarity calculation (yellow bars are the diffusion input vertices, and their contribution is minimized by the algorithm as per the choice $k = 1$, see Section 3.3.1) between neurons. Note that meaningful contributions to similarity can be made by neurons that are not directly connected to AVB or AVE pairs.

Closer views of the in-neighbor and out-neighbor subgraphs of A[i, j] reveal details of the diffusion profile vectors as seen in Figure 8. For the AVB pair, for example, diffusion flow between AVBL and RIFL and also between AVBR and RIFR is large and asymmetric, reflecting neuron laterality (Figure 8(a)). The contribution of this vertex to similarity between AVBL and AVBR is scaled by diffusion flow to the contralateral neuron (between AVBL and RIFR and between

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

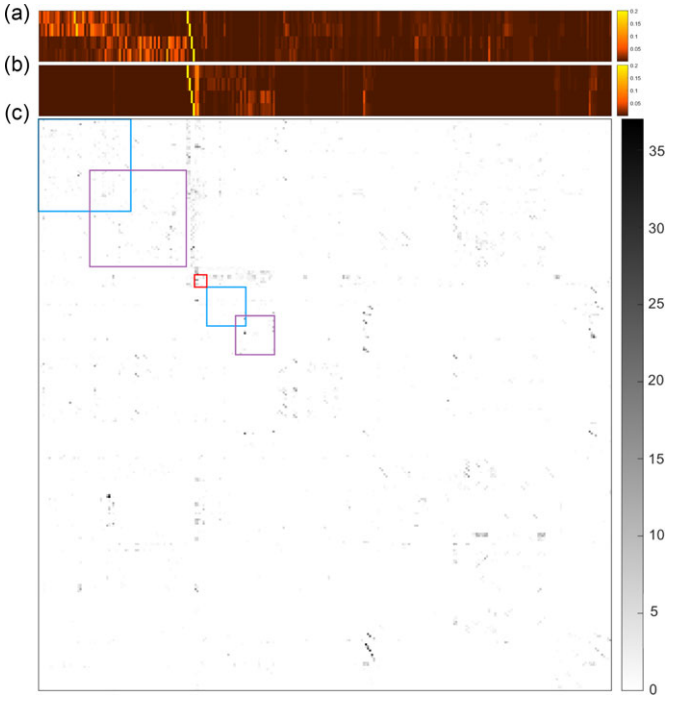

Figure 7. Top two pictures: Diffusion profile vectors used to calculate similarity between neurons. Each row is the vector for each of the four neurons in [AVB, AVE] pairs. a. In-flow diffusion profiles. Yellow bars indicate diffusion injection sites. b. Out-flow diffusion profiles. c. Sorted chemical synapse adjacency matrix displaying weights as gray scale. Sorting for all panels proceeds in order of [AVB, AVE] in-neighbors (large blue and purple boxes, respectively), the four neurons AVBL, AVBR, AVEL, AVER , neurons that are both in- and out-neighbors (red box), and neurons that are [AVB, AVE] out-neighbors.

AVBR and RIFL) and is matched by common diffusion flow into other vertices (e.g., AVM) from both AVBL and AVBR. AVM is a low threshold touch receptor that projects broadly to command interneurons for forward and backward movement (Chalfie & Sulston, 1981; Leifer et al., 2011; Pirri & Alkema, 2012). Within the AVE pair, contributions to similarity also arise from asymmetric (RIGL and RIGR) and common diffusion flows (e.g., ALA and RIS). ALA is a high threshold mechanoreceptor, induces lethargy, and inhibits the AVE pair (Sanders et al., 2013; Van Buskirk & Sternberg, 2007; Altun et al., *n.d.*). ALA and AVM are representative of unpaired neurons that can interact with both elements of paired neurons, (in the case of ALA via bifurcating processes) and multiple pairs, and thereby contribute to similarity measures within pairs. Parallel diffusion flow patterns can be identified in the out-flow diffusion vectors (Figure 8(b)).

Diffusion similarity between the elements of the [AVB] pair and the [AVE] pair derive largely from shared inputs, and these can be contralaterally (e.g., in-neighbor AVJR diffusion flow from AVBL and AVER, Figure 8(a)) or ipsilaterally matched (e.g., out-neighbor VA02 diffusion flow from AVBL and AVEL). The vertices most consistently linked to all four neurons of the [AVB, AVE] pair are the [AVAL, AVAR] pair, which not surprisingly, because of their role as command interneurons for locomotion (Chalfie et al., 1985), are both in- and out-neighbors (red box in Figures 7(c) and 8(b)). Importantly, other contributions to between pair similarity arise from vertices not directly connected to any of these four neurons (see Figure 7).

The diffusion similarity matrices quantify these observations of features of the diffusion profile vectors (Figure 8(c) and (d)). Note the strong similarity within [AVB, AVE] pairs that is matched in value among the in-neighbors [RIML, RIMR] and [AVAL, AVAR] pairs (Figure 8(c)) as well as among the out-neighbors [SABVL, SABVR], [DA03, DA04] and [ASA07, ASA09) motor neuron

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

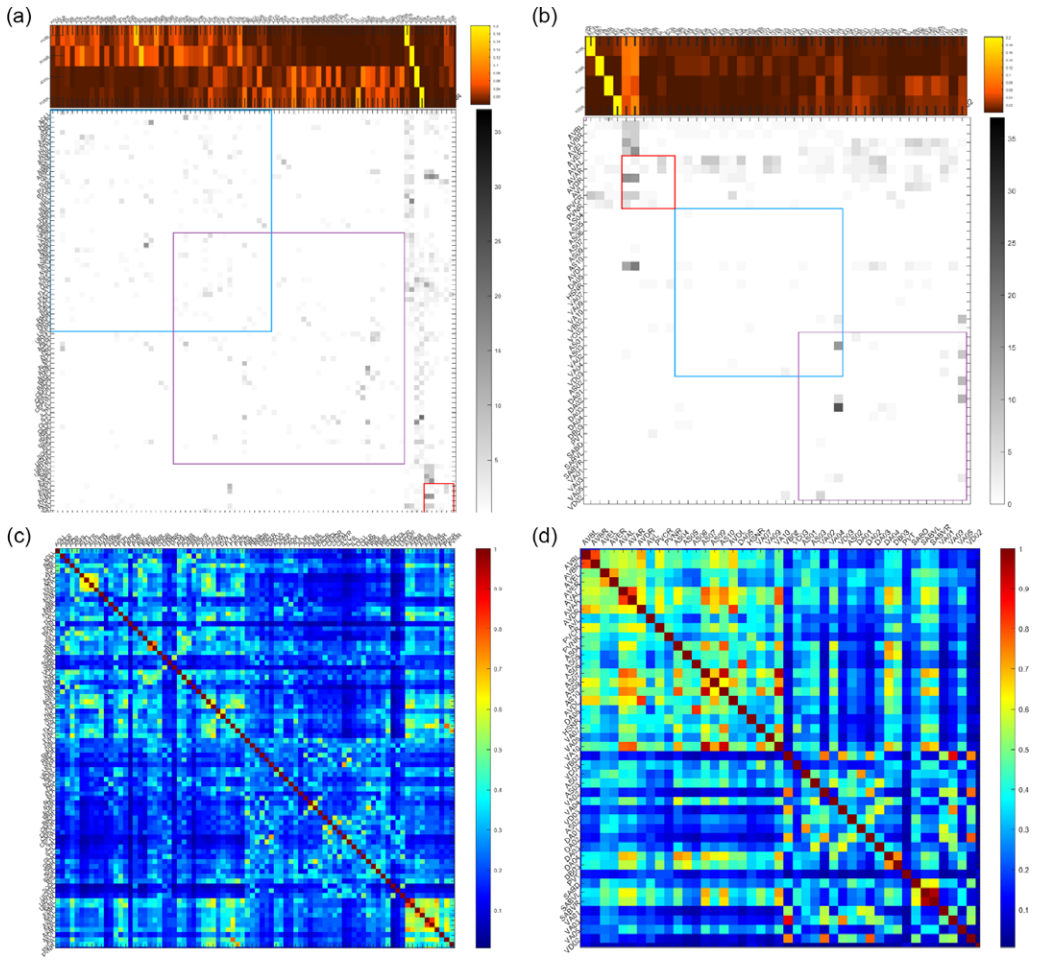

Figure 8. Zoomed in views of Figure 7, showing a. in-neighbor and b. out-neighbor components of the diffusion vectors and subsets of the induced adjacency matrix. Note the relative lack of connectivity within and between [AVB AVE] pairs. c. Corresponding submatrix of the similarity matrix S to the induced adjacency matrices in panel a. d. Similar correspondence to panel b.

pairs (Figure 8(d)). The [AVB, AVE] pairs show strong similarity to each other and also to the [AVA] pair, which coincides with their associated functions as command interneurons mediating locomotion. Dynamic analysis of population neural activity during behavioral pirouettes (short reversal, change in, and resumption of original direction) showed synchronized [ABA, AVE] pair activity and anticorrelated [AVB] pair activity consistent with their original association of movement direction through lesion studies (Kato et al., 2015; Chalfie et al., 1985). Although the [AVA] pair, like the [AVE] pair, functions in controlling backward movement, the [AVA, AVB] pairs, but not the [AVE] pair, extend the length of the ventral cord. This similar structure appears to be reflected in the slightly higher similarity index between the [AVA, AVB] pairs than between the [AVA, AVE] pairs (see Figures 8(c) and (d)).

Neurons outside of those directly projecting to any of the [AVB, AVE] pairs (see Figure 7) can contribute to similarity, primarily within pairs. Among the vertices contributing to the in-flow diffusion pattern for the [AVB,AVE] pairs, these can be sensory, as in the mechanosensitive neurons [PDEL, PDER] that mediate locomotion based on fed state (Sawin et al., 2000), interneurons [SAADL, SAADR], or motor, as in the ring motorneuron RMED which innervates dorsal muscles

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

of the head (White et al., 1986; Cook et al., 2019) and HSNL, which contributes to coordination of egg-laying and locomotion (Schafer, 2005; Eisenmann, 2005). Diffusion out-flow contributions include, not surprisingly, dorsal and ventral inhibitory neurons [DD, VD], which are key elements of an integrated motor control network for locomotion (Schuske et al., 2004).

The application of the diffusion method provides a framework for the exploration of the quantitative relationships between neurons and the larger scale relationships among neuron pairs and groups. Insights into indirect neuron interactions can be gained by studying diffusion profile vectors. But moreover, the similarity measure is able to effectively capture both similarities and differences between the AVB and AVE pairs studied here.

5.2 Similarity within the inner plexiform layer of the mouse retina

In order to test our similarity method in a real-world dataset of larger size and complexity, we selected the mouse retinal neuron connectome data known as e2006 (Briggman et al., 2011; Helmstaedter et al., 2013). The dataset includes a weighted undirected network on 903 neurons (bipolar, horizontal, amacrine, and ganglion cells) and 173 glial cells. In that network, the edge weight between neuron i and j is the area of contact (μm^2) between neuron i and j which was revealed to be a good predictor of the existence and weight of synapses, although imaging did not permit description as a directed network. For our analysis of the behavior of our similarity method with respect to organized structure in the mouse retina network, we adopt a ground truth approach. We propose that certain groups of neurons may be treated as modules of functionally relevant connectivity based on their known connectivity patterns and well-studied biological function. We then verify that our similarity measure reflects these modules in a quantitative way.

Neuron to neuron synaptic activity in the mouse retina takes place in neuropil regions known as the outer and inner plexiform layers (OPL and IPL) (Kolb, 2011). We focus on the processes of bipolar, amacrine, and ganglion cells that make contact primarily in the IPL. In particular, we investigated the neural circuits driven by rod photoreceptors to map visual space under low light conditions (scotopic vision). A narrow field amacrine cell, called A2, plays a key role in this process by collecting input from rod bipolar cells (mostly ON type defined by activity in light) and must suppress activity in cone OFF bipolar cells, which are more active in decreasing light intensity (Marc et al., 2014). The ON and OFF sublaminae occupy the inner and outer layers of the IPL, respectively.

A2 cell arbors form a tiling across the retinal plane, hence each A2 cell dominates a local region of rod and OFF cone bipolar cells. It is this expectation of translation invariant "functional modules" (Jonas & Kording, 2015) that motivates our approach here to use local modules of A2, OFF cone bipolar, and rod bipolar cells as a ground truth for modular structure that would imply vertex similarity within that structure. That is, each of the 35 A2 cells and their associated modules in the dataset should have relatively higher similarity within module than between modules.

In order to test this hypothesis, we first computed the $1,076 \times 1,076$ similarity matrix S from the weighted adjacency data of 1,076 neuronal and glial cells. We assembled ground truth modules based around A2 cells as follows. For visual simplicity, we restricted our study of OFF cone bipolar cells to the type CBC1 as it has the largest amount of connectivity with A2 in this dataset. This together with A2 and rod bipolar cells yielded a subgroup of 195 neurons. Each CBC1 and rod bipolar cell was grouped with the A2 cell with which it has the highest contact area, so each module has the same internal connectivity motif. A view looking "down" at the retinal plane illustrates the modules by colors and numeric labels (Figure 9(f)). Four examples of the modules seen with their neuron processes projecting down through the OFF and ON sublamina of the IPL are shown in Figure 9(e) (OFF and ON sublaminae are bottom two layers). The adjacency and similarity submatrices have the neurons ordered identically (Figure 9(c) and (d)), revealing the connectivity motif of dominance by local A2 cells. Note that CBC1 and rod cells can have contact with more than one A2 cell and so there is connectivity between modules governed by proximity within the

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

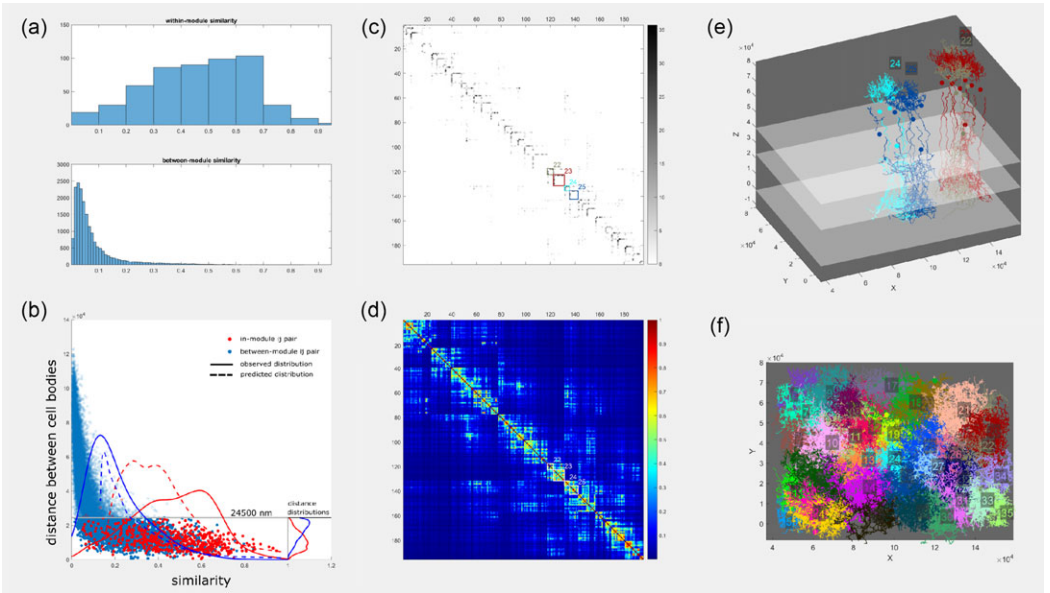

Figure 9. a. Histograms of similarity values over pairs of neurons within modules and pairs of neurons between modules, respectively b. Distance in nanometers in the *XY* (retinal) plane between neuron i and j plotted against their similarity, blue are ij pairs where the neurons are in different modules, red are ij pairs of neurons in the same module, observations above 24, 500 nm are shown translucent, and the overlay pdfs are in units of probability density c. Adjacency data of the 195 neurons of interest, the ordering is module 1, module 2, . . . and so on, within each module the first neuron is the A2, followed by CBC1 followed by rod d. Similarity matrix of the 195 neurons of interest, the ordering matched the adjacency data above it e. Side view of four example modules indicated by color, the colors and labels match the annotated subsets of the adjacency data seen to the left f. view looking "down" through the retinal plane, colors and numeric labels indicate the modules and match those in Figures c,d, and e.

retinal plane. The cells examined in these submatrices can have connectivity with other cells in the dataset, which likely influences the cell to cell similarity values we computed. Nonetheless, the extensive mutual contact within this submatrix and also their known biological functional relationships are likely driving factors in the high similarity according to our calculation S.

The adjacency matrix reveals intra-module and, for some modules, between-module connectivity (example pairs 22–23 and 24–25) that is reflected in similarity between modules 22 and 23 but not between either of these modules and modules 24 and 25 (Figure 9(c) and (d)). These module pairs are each neighbors with some shared connectivity (Figure 9(c), (e), and (f)) and illustrate the tendency of neurons to be less similar with distance across the retinal sheet and more similar when they are neighbors. Basic statistics of the within-module and between-module distributions of similarity reveal that similarity within modules tends higher ($46 \pm 16\%$ (sd)) than between-module pairs ($8 \pm 9\%$ (sd)), representing a correlation between our similarity measure and the ground truth modules. Histograms of these distributions are seen in Figure 9(a). We computed the uncertainty score (Section 4.2) of the similarity with respect to the ground truth modules and found it to be 0.11, a value indicating that the modules are fairly well represented as internally similar and less similar between modules.

The decrease in similarity with distance between cell bodies was best fit by a degree one rational function $S = \frac{3646}{d+3035}$ ($r^2 = 0.6056$; Figure 9(b)) of the distance d between neurons. The tail of this distribution reveals that all observed similarities (except for one outlier) of at least 50% are between pairs at a proximity of 24.5 μm or less, a distance which excludes only six in-module pairs. This distance is slightly larger than the diameter of modules when viewed in the retinal plane (Figure 9(f)), reflecting high similarity within modules or between adjacent modules.

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

We next focused on pairs closer than the 24.5 μm threshold and used the kernel density estimator method to compute pdfs separately for in-module observations and between-module observations of both similarity and distance (Figure 9(b), solid red and blue lines, respectively, in lower right box aligned with the distance axis). We then inferred distributions of similarity (red and blue dashed lines, respectively, in Figure 9(b)) that would be expected based on the observed distributions of distance given the rational function fit relating similarity and distance. The observed between-module distribution was similar in shape and peak location, although broader than the predicted distribution (solid and dashed blue lines, respectively, in Figure 9(b)). In contrast, the observed in-module similarities have a significant rightward skew relative to the predicted distribution (peak value 60% vs prominent modes at about 25% and 40%). It is in that sense that the within-module similarity values reflect the property of being in the same functional module beyond what can be accounted for by close proximity and seem to be distinguished from between-module similarity values. In this way, the similarity matrix derived from the adjacency data for this volume of neurons in the mouse retina can be used to identify modular structures. This can be used to either confirm existing biological information or to reveal unknown structures such as members of these modules that may not have been previously known or repetition of these modules in other areas of the connectome once these data are available.

6. Conclusions

In this work, we have presented a method for developing a measure of similarity between network vertices derived from the patterns of diffusion found in the network in which they are situated. The method can be applied to a variety of graphical structures, and the resulting similarity matrix reveals not only similarity properties within the local neighborhood of a vertex but also identifies collections of similar vertices that exist in parallel or layered structures within the network such as those found in neural connectomes. We applied this methodology to the study of a number of synthetic networks and demonstrate its ability to identify communities of vertices as well as parallel substructures. We then applied the process to the study of the *C.elegans* connectome and identified parallel information processing networks within that connectome that share similar properties but that are not directly interconnected. We then performed the same analysis on a section of the IPL of the mouse retina and again identified similar structures based on this vertex similarity. This type of structure detection is essential to the study of large, complex networks like connectomes, since higher-order neural functions such as stimulus response rely on parallel processing pathways to filter and distinguish different phenomena. Future work will use this similarity data as the basis for a community detection algorithm, and we will apply this combination to enhance the discovery of biological functional structures. Specifically, since intra-module functional relations may be better represented in the similarity matrix, community structures related to this similarity data would in principle collect these modular structures in communities and allow for the efficient detection of these structures in the large networks being identified by ongoing nanoscale connectomics research. For these applications, where future datasets will contain many more nodes and have greater incidence of long distance connectivity, community detection methodologies based on a similarity measure as presented here could be used to identify either unknown structures or to verify the existence of known processes, such as the modules found in the mouse retina, in other areas of the neural structures being studied. Another potential application will be to the validation of some portions of the synaptic connectivity derived from new image volumes in this work as expected connectivities can be compared to those observed once the patterns have been identified in validated datasets. More generally, in combination with efficient community detection algorithms, this work can be applied in a variety of other contexts as well to identify structure in large, complex networks in ways that complement existing methods.

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

Funding. This work was supported in part by NIH Grant R01 DC015901 and NSF Grant DMS-1700218.

Conflicts of interest. None.

References

Altun, Z., Hall, D. H., Wolkow, C. A., Crocker, C., & Lints, R. *Handbook of C. Elegans Anatomy. Wormatlas 2002–2021*.

Angstmann, C. N., Donnelly, I. C., & Henry, B. I. (2013). Pattern formation on networks with reactions: A continuous-time random-walk approach. *Physical Review E*, *87*(Mar), 032804.

Avrachenkov, K., Chebotarev, P., & Rubanov, D. (2019). Similarities on graphs: Kernels versus proximity measures. *European Journal of Combinatorics*, *80*, 47–56. Special Issue in Memory of Michel Marie Deza.

Bai, X., Wilson, R. C., & Hancock, E. R. (2005). Manifold embedding of graphs using the heat kernel. In R. Martin, H. Bez, & M. Sabin (Eds.), *Mathematics of surfaces xi* (pp. 34–49). Berlin, Heidelberg: Springer Berlin Heidelberg.

Bock, D. D., Lee, W.-C. A., Kerlin, A. M., Andermann, M. L., Hood, G., Wetzel, A. W., Yurgenson, S., Soucy, E. R., Kim, H. Suk, & Reid, R. C. (2011). Network anatomy and in vivo physiology of visual cortical neurons. *Nature*, *471*(7337), 177–182.

Bondy, A., & Murty, M. R. (2008). *Graph theory*. Springer.

Brandes, U. (2016). Network positions. *Methodological Innovations*, *9*, 1–19.

Brauer, F., & Nohel, J. (1969). *The qualitative theory of ordinary differential equations, an introduction*. New York: W. A. Benjamin.

Brenner, S. (1973). The Genetics of Behaviour. *British Medical Bulletin*, *29*(3), 269–271.

Briggman, K. L., Helmstaedter, M., & Denk, W. (2011). Wiring specificity in the direction-selectivity circuit of the retina. *Nature*, *471*(7337), 183–188.

Chalfie, M., & Sulston, J. (1981). Developmental genetics of the mechanosensory neurons of caenorhabditis elegans. *Developmental Biology*, *82*(2), 358–370.

Chalfie, M, Sulston, J. E., White, J. G., Southgate, E., Thomson, J. N., & Brenner, S. (1985). The neural circuit for touch sensitivity in Caenorhabditis Elegans. *Journal of Neuroscience*, *5*(4), 956–964.

Chan, P. K., Schlag, M. D. F., & Zien, J. Y. (1994). Spectral k-way ratio-cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *13*(9), 1088–1096.

Chen, B. L., Hall, D. H., & Chklovskii, D. B. (2006). Wiring optimization can relate neuronal structure and function. *Proceedings of the National Academy of Sciences*, *103*(12), 4723–4728.

Chen, B. L.-J. (2007). *Neuronal network of c.elegans: From anatomy to behavior*. Ph.D. thesis, The Watson School of Biological Sciences at Cold Spring Harbor Laboratory.

Cheng, X., Rachh, M., & Steinerberger, S. (2019). On the diffusion geometry of graph Laplacians and applications. *Applied and Computational Harmonic Analysis*, *46*(3), 674 – 688.

Chung, F. (1997). *Spectral graph theory*. American Mathematical Society.

Chung, F. (2007). The heat Kernel as the pagerank of a graph. *Proceedings of the National Academy of Sciences*, *104*(50), 19735–19740.

Cook, S. J., Jarrell, T. A., Brittin, C. A., Wang, Y., Bloniarz, A. E., Yakovlev, M. A., Nguyen, K. C. Q., Tang, L. T. H, Bayer, E. A., Duerr, J. S., Bülow, H. E., Hobert, O., Hall, D. H., & Emmons, S. W. (2019). Whole-animal connectomes of both caenorhabditis elegans sexes. *Nature*, *571*(7763), 63–71.

Cooper, K., & Barahona, M. (2010). *Role-based similarity in directed networks*.

Corsi, A. K, Wightman, B., & Chalfie, M. (2015). A transparent window into biology: A primer on caenorhabditis elegans. *Genetics*, *200*(2), 387–407.

Delvenne, J.-C., Schaub, M.T., Yaliraki, S.N., & Barahona, M. (2013). The stability of a graph partition: A dynamics-based framework for community detection. *Dynamics On and Of complex Networks*, *2*, 221–242.

Diestel, R. (2017). *Graph theory*. Springer.

Eisenmann, D. M. (2005). *Wormbook: The online review of c. elegans biology*. Research Community, Wormbook.

Emmons, S. W. (2015). The beginning of connectomics: A commentary on White et al. (1986) The structure of the nervous system of the nematode Caenorhabditis elegans. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *370*(1666), 20140309.

Erdős, P., & Rényi, A. (1959). On random graphs I. *Publ. Math. Debrecen*, *6*, 290–297.

Erdős, P., & Rényi, A. (1960). On the evolution of random graphs. *Publ Math. Inst*, *5*, 17–61.

Estrada, E., & Silver, G. (2017). Accounting for the role of long walks on networks via a new matrix function. *J. Appl Math. Anal*, *449*, 1581–1600.

Fiedler, M. (1989). Laplacian of graphs and algebraic connectivity. *Banach Center Publications*, *25*(1), 57–70.

Fouss, F., Yen, L., Pirotte, A., & Saerens, M. (2006). An experimental investigation of graph kernels on a collaborative recommendation task. In *Proceedings of the Sixth International Conference on Data Mining, ICDM'06* (pp. 863–868). IEEE.

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

Fouss, F., Pirotte, A., Renders, J.-M., & Saerens, M. (2007). Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *19*(3), 355–369.

Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points. *Science*, *315*(5814), 16.

Frieze, A., & Karoński, M. (2016). *Introduction to random graphs*. Cambridge University Press.

Genton, M. G. (2002). Classes of Kernels for machine learning: A statistics perspective. *Journal of Machine Learning Research*, *2*(Mar.), 299–312.

Ghawalby, H. E., & Hancock, E. R. (2015). Heat Kernel embeddings, differential geometry and graph structure. *Axioms*, *4*, 275–293.

Gray, J. M., Hill, J. J., & Bargmann, C. I. (2005). A circuit for navigation in caenorhabditis elegans. *Proceedings of the National Academy of Sciences*, *102*(9), 3184–3191.

Helmstaedter, M., Briggman, K. L, Turaga, S. C, Jain, Viren, S., Sebastian, H. & Denk, W. (2013). Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature*, *500*(7461), 168–174.

Huang, W., Segarra, S., & Ribeiro, A. (2015). Diffusion distance for signals supported on networks. *2015 49th asilomar conference on signals, systems and computers* (pp. 1219–1223).

Jabr, F. (2012). The connectome debate: Is mapping the mind of a worm worth it? *Scientific American*, *18*.

Jaccard, P. (1901). Distribution de la flore alpine dans le bassin des dranses et dans quelques regions voisines. *Bulletin de la société vaudoise des sciences naturelles*, *37*, 241–272.

Jonas, E., & Kording, K. (2015). Automatic discovery of cell types and microcircuitry from neural connectomics. *Elife*, *4*, e04250.

Kato, S., Kaplan, H. S., Schrödel, T., Skora, S., Lindsay, T. H., Yemini, E., Lockery, S., & Zimmer, M. (2015). Global brain dynamics embed the motor command sequence of caenorhabditis elegans. *Cell*, *163*(3), 656–669.

Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika*, *18*(1), 39–43.

Kolb, Helga. (2011). Inner plexiform layer. InH. Kolb, R. Nelson, E. Fernandez, & B. Jones (Eds.), *Webvision: The organization of the retina and visual system*. University of Utah Health Sciences Center.

Kondor, R. I., & Lafferty, J. (2002). Diffusion kernels on graphs and other discrete input spaces. In*Proceedings of ICML* (pp. 315–322).

Kovács, I. A., Luck, K., Spirohn, K., Wang, Y., Pollis, C., Schlabach, S., Bian, W., Kim, D.-K., Kishore, N., & Hao, T. (2019). Network-based prediction of protein interactions. *Nature Communications*, *10*(1), 1–8.

Leicht, E. A., Holme, P., & Newman, M. E. J. (2006). Vertex similarity in networks. *Physics Review E*, *73*.

Leifer, A. M., Fang-Yen, C., Gershow, M., Alkema, M. J., & Samuel, A. D. T. (2011). Optogenetic manipulation of neural activity in freely moving caenorhabditis elegans. *Nature Methods*, *8*(2), 147–152.

Lenart, C. (1998). A generalized distance in graphs and centered partitions. SIAM Journal on Discrete Mathematics, *11*(2), 293–304.

Liben-Nowell, D., & Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the Association for Information Science and Technology*, *58*(7), 1019–1031.

Lovász, L. (1993). Random walks on graphs: A survey. *Bolyai Society Mathematical Studies: Combinatorics - paul erdös is eighty*, *2*, 1–46.

Lü, L., & Zhou, T. (2011). Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and Its Applications*, *390*(6), 1150–1170.

Lü, L., Zhang, Y.-C., Yeung, C. H., & Zhou, T. (2011). Leaders in social networks, the delicious case. *Plos One*, *6*(6). e21202.

Luo, D., Ding, C., Huang, H., & Li, T. (2009). Non-negative Laplacian embedding. In *2009 Ninth IEEE International Conference on Data Mining* (pp. 337–346). IEEE.

Marc, R. E., Anderson, J. R., Jones, B. W., Sigulinsky, C. L., & Lauritzen, J. S. (2014). The AII amacrine cell connectome: A dense network hub. *Frontiers in Neural Circuits*, *8*, 104.

Masuda, N., Porter, M. A., & Lambiotte, R. (2017). Random walks and diffusion on networks. *Physics Reports*, *716–717*(November), 1–58.

Meila, M., & Shi, J. (2001). A random walks view of spectral segmentation. *Proceedings of the 8th international workshop on artificial intelligence and statistics*.

Mohar, B. (1989). Isoperimetric numbers of graphs. *Journal of Combinatorial Theory, Series B*, *47*(3), 274 – 291.

Ohyama, T., Schneider-Mizell, C. M., Fetter, R. D., Aleman, J. V., Franconville, R., Rivera-Alba, M., . . . Zlatic, M. (2015). A multilevel multimodal circuit enhances action selection in drosophila. *Nature*, *520*, 633–639.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The pagerank citation ranking: Bringing order to the web*. Technical Report, Stanford InfoLab.

Pech, R., Hao, D., Lee, Y.-L., Yan, Y., & Zhou, T. (2019). Link prediction via linear optimization. *Physica A: Statistical Mechanics and Its Applications*, *528*, 121319.

Perrault-Joncas, D. C., & Meila, M. (2011). Directed graph embedding: An algorithm based on continuous limits of Laplacian-type operators. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 24* (pp. 990–998). Curran Associates, Inc.

https://doi.org/10.1017/nws.2021.11 Published online by Cambridge University Press

Petit, J., Lambiotte, R., & Carletti, T. (2019). Classes of random walks on temporal networks with competing timescales. *Applied Network Science*, *4*(1), 72.
Pirri, J. K., & Alkema, M. J. (2012). The neuroethology of c. elegans escape. *Current Opinion in Neurobiology*, *22*(2), 187 – 193.
Qi, X., Wu, Q., Zhang, Y., Fuller, E., & Zhang, C.-Q.. (2011). A novel model for dna sequence similarity analysis based on graph theory. *Evolutionary Bioinformatics*, *7*, EBO–S7364.
Qi, X., Duval, R. D., Christensen, K., Fuller, E., Spahiu, A., Wu, Q., . . . Zhang, C. (2013). Terrorist networks, network energy and node removal: A new measure of centrality based on Laplacian energy. *Social Networking*, *2*(01), 19.
Rosvall, M., & Bergstrom, C. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, *105*(4), 1118–1123.
Rosvall, M., Axelsson, D., & Bergstrom, C. (2009). The map equation. *European Physical Journal Special Topics*, *178*, 13–23.
Ryan, K., Lu, Z., & Meinertzhagen, I. A. (2016). The CNS connectome of a tadpole larva of *Ciona intestinalis* (l.) highlights sidedness in the brain of a chordate sibling. *elife*, *5*(dec.), e16962.
Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval*. McGraw-Hill computer science series. McGraw-Hill.
Sanders, J., Nagy, S., Fetterman, G., Wright, C., Treinin, M., & Biron, D. (2013). The caenorhabditis elegans interneuron ALA is (also) a high-threshold mechanosensor. *BMC Neuroscience*, *14*(Dec), 156–156.
Sawin, E. R., Ranganathan, R., & Horvitz, H. R. (2000). C. elegans locomotory rate is modulated by the environment through a dopaminergic pathway and by experience through a serotonergic pathway. *Neuron*, *26*(3), 619–631.
Schafer, W. R. (2005). Deciphering the neural and molecular mechanisms of c. elegans behavior. *Current Biology*, *15*(17), R723–R729.
Schuske, K., Beg, A. A., & Jorgensen, E. M. (2004). The GABA nervous system in c. elegans. *Trends in Neurosciences*, *27*(7), 407–414.
Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge Univ Press.
Shi, J., & M., J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine intelligence*, *22*(8), 888–905.
Shirkhorshidi, A. S., Aghabozorgi, S., & Wah, T. Y. (2015). A comparison study on similarity and dissimilarity measures in clustering continuous data. *PLOS ONE*, *10*(12), 1–20.
Smola, A. J., & Kondor, R. (2003). Kernels and regularization on graphs. In *Learning theory and kernel machines* (pp. 144–158). Springer.
Takemura, S.-y, Nern, A., Chklovskii, D. B., Scheffer, L. K., Rubin, G. M., & Meinertzhagen, I. A. (2017). The comprehensive connectome of a neural substrate for on motion detection in *Drosophila*. *elife*, *6*(Apr), e24394.
Thiel, K., & Berthold, M. R. (2010). Node similarities from spreading activation. In *2010 IEEE international conference on data mining* (pp. 1085–1090).
Towlson, E. K., Vértes, P. E., Ahnert, S. E., Schafer, W. R., & Bullmore, E. T. (2013). The rich club of the c.elegans neuronal connectome. *Journal of Neuroscience*, *10*(33), 15.
Van Buskirk, C., & Sternberg, P. W. (2007). Epidermal growth factor signaling induces behavioral quiescence in caenorhabditis elegans. *Nature Neuroscience*, *10*(10), 1300–1307.
Varshney, L. R., Chen, B. L., Paniagua, E., Hall, D. H., & Chklovskii, D. B. *Neuronal connectivity II*. http://www.wormatlas.org/neuronalwiring.html.
Varshney, L. R., Chen, B. L., Paniagua, E., Hall, D. H., & Chklovskii, D. B. (2011). Structural properties of the caenorhabditis elegans neuronal network. *Plos Computational Biology*, *7*(2), e1001066.
West, D. (2001). *Introduction to graph theory*. Prentice Hall.
White, J. G. (2013). Getting into the mind of a worm–a personal view. *Wormbook*, 1–10.
White, J. G., Southgate, E., Thomson, J. N., & Brenner, S. (1986). The structure of the nervous system of the nematode caenorhabditis elegans. *Philosophical Transactions of the Royal Society of London Series B, Biological Sciences*, *314*(1165), 1–340.
Zheng, Z., Lauritzen, J. S., Perlman, E., Robinson, C. G., Nichols, M., Milkie, D., . . . Bock, D. D. (2018). A complete electron microscopy volume of the brain of adult drosophila melanogaster. *Cell*, *174*(3), 730–743.e22.
Zhou, T., Lü, L., & Zhang, Y.-C. (2009). Predicting missing links via local information. *The European Physical Journal B*, *71*(4), 623–630.
Zhou, T., Lee, Y.-L, & Wang, G. (2021). Experimental analyses on 2-hop-based and 3-hop-based link prediction algorithms. *Physica A: Statistical Mechanics and Its Applications*, *564*. Article 125532.

Cite this article: Payne S., Fuller E., Spirou G. and Zhang C.-Q. (2021). Diffusion profile embedding as a basis for graph vertex similarity. *Network Science* **9**, 328–353. https://doi.org/10.1017/nws.2021.11