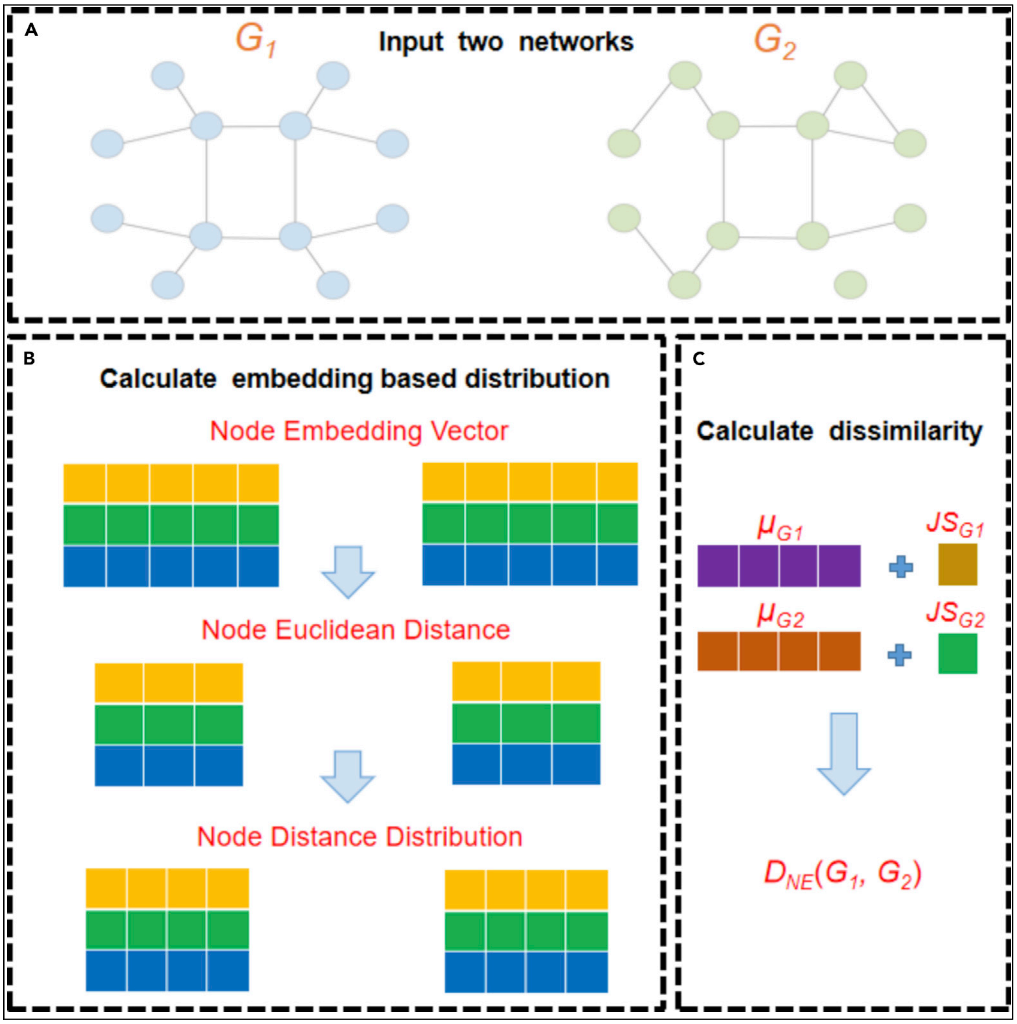
iScience

CellPress
OPEN ACCESS

Article

Quantification of network structural dissimilarities based on network embedding



Zhipeng Wang,
Xiu-Xiu Zhan,
Chuang Liu, Zi-Ke
Zhang

zhanxiuxiu@hznu.edu.cn (X.-X.Z.)
zkz@zju.edu.cn (Z.-K.Z.)

Highlights

Capture global structural information of any given network

Superior to various baselines in both synthetic and real-world networks

Applicable to compare networks with different sizes and types

Wang et al., iScience *25*, 104446
June 17, 2022 © 2022 The Author(s).
https://doi.org/10.1016/j.isci.2022.104446

iScience

CellPress
OPEN ACCESS

Article

Quantification of network structural dissimilarities based on network embedding

Zhipeng Wang,[1] Xiu-Xiu Zhan,[1,*] Chuang Liu,[1] and Zi-Ke Zhang[2,*]

SUMMARY

Quantifying structural dissimilarities between networks is a fundamental and challenging problem in network science. Previous network comparison methods are based on the structural features, such as the length of shortest path and degree, which only contain part of the topological information. Therefore, we propose an efficient network comparison method based on network embedding, which considers the global structural information. In detail, we first construct a distance matrix for each network based on the distances between node embedding vectors derived from *DeepWalk*. Then, we define the dissimilarity between two networks based on Jensen-Shannon divergence of the distance distributions. Experiments on both synthetic and empirical networks show that our method outperforms the baseline methods and can distinguish networks well. In addition, we show that our method can capture network properties, e.g., average shortest path length and link density. Moreover, the experiment of modularity further implies the functionality of our method.

INTRODUCTION

Network is a natural representation of complex data associations and it has been used in many domains ranging from biology (Liu et al., 2020) and physics (Boccaletti et al., 2006) up to social sciences (Strogatz, 2001). Because of the specific characteristics of the complex system it represents, network emerges complex non-trivial topological features, such as scale-free (Barabási and Albert, 1999) and small-world properties (Watts and Strogatz, 1998). The flexibility of network modeling and the rapid growth of network data in recent years make it urgent to design effective network comparison methods. Because comparing structural similarities between networks is an important task, which has various scientific applications, e.g., the comparison of brain networks for different subjects (Bullmore and Sporns, 2009) and diffusion cascade of news (Zhan et al., 2018), the classification of proteins (Liu et al., 2020), the identification of changing points of temporal networks (Holme and Saramäki, 2012), and the evaluation of generative network models (Hartle et al., 2020; Ali et al., 2014; De Domenico et al., 2015).

Researchers have proposed methods based on graph isomorphism to compare networks (Zemlyachenko et al., 1985; Babai, 2016; Grohe and Schweitzer, 2020). The main limitations of isomorphism-based methods are as follows: first of all, isomorphism-based methods can only compare networks with the same size and are not scalable to large networks with millions of nodes. Secondly, this kind of methods can only tell whether two networks are isomorphic or not but to what extent two networks are different is hardly measured. Thanks to the mature research of network topology mining (Costa et al., 2007; Martínez and Chavez, 2019; Tsitsulin et al., 2018; Gärtner et al., 2003), a number of researchers have studied how to use network characteristics, e.g., adjacency matrix, node degree, and shortest path distance, to compare networks with huge and different sizes. For instance, Saxena et al. (2019) introduced a network similarity method based on hierarchical diagram decomposition via using Canberra distance, which considers both local and global network topology. Lu et al. (2014) proposed a manifold diffusion method based on random walk, which can not only distinguish networks with different degree distributions but also can discriminate networks with the same degree distribution. Beyond the direct comparison of network topology, we have witnessed the effectiveness of using quantum information science, i.e., information entropy, in network comparison. For example, De Domenico and Biamonte (2016) proposed a set of information theory tools for network comparison based on spectral entropy. Schieber et al. (2017) quantified the dissimilarities between networks by considering the probability distribution of the shortest path distance between nodes. Chen et al. (2018) proposed a comparison method based on the node communicability

[1] Research Center for Complexity Sciences, Hangzhou Normal University, Hangzhou 311121, PR China

[2] College of Media and International Culture, Zhejiang University, Hangzhou 310058, PR China

*Correspondence: zhanxiuxiu@hznu.edu.cn (X.-X.Z.), zkz@zju.edu.cn (Z.-K.Z.)

https://doi.org/10.1016/j.isci.2022.104446

sequence entropy. Bagrow and Bollt (2019) proposed a method based on portrait divergence to compare networks. The portrait divergence-based method incorporates the topological characteristics of networks at all scales and is applicable to all types of networks. The basic idea behind this kind of methods is that one specific network property, such as the shortest path distance (Schieber et al., 2017) and node communicability matrix (Chen et al., 2018), is chosen to measure the information content of a network via a proper entropy. Therefore, the dissimilarity between two networks is given by the difference between the information content of them. However, we claim that the selection of one specific property as a representative of network information content may not be able to capture the information of a whole network. For example, we can quantify the network dissimilarities through comparing the distance distribution based on the information entropy. However, the shortest path-based distance between nodes is only one kind of properties in a network; it cannot represent the complete structure of a network. Therefore, how to extract network features sufficiently to quantify network differences is an urgent problem to be solved.

Network embedding, which aims to embed each node into a low-dimensional vector by preserving the network structure as much as possible, has been widely used to solve many problems in network science, e.g., link prediction (Bu et al., 2019; Grover and Leskovec, 2016), community detection (Jin et al., 2019; Li et al., 2016; Fortunato, 2010), and network reconstruction (Pio et al., 2020; Xu et al., 2020; Goyal and Ferrara, 2018). In this paper, we further widen the application of network embedding, i.e., we explore how to use network embedding to characterize the dissimilarity of two networks in a state-of-the-art way. We start from using a simple and fast network embedding algorithm, i.e., *DeepWalk*, which can capture the global information of a network, to measure the distance between two nodes for a given network. Then, the information content of a network, i.e., network similarity heterogeneity, is defined based on the node distance distribution and Jensen-Shannon divergence. Accordingly, the dissimilarity between two networks is further defined upon network similarity heterogeneity between a pair of networks. We validate the effectiveness of the network embedding-based comparison method on both synthetic and empirical networks. Compared to the baseline methods, network embedding-based comparison shows high distinguishability.

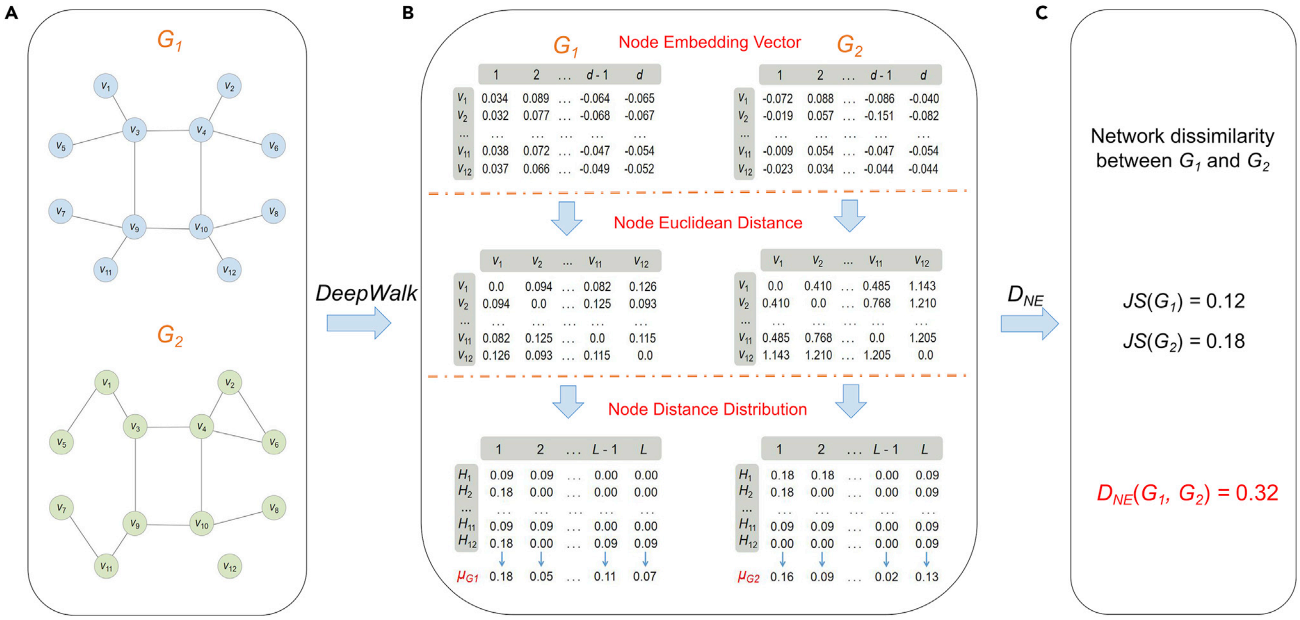
RESULTS

Embedding-based network dissimilarity

Given a network $G = (V, E)$, in which V represents the node set, and $E = \{(v_i, v_j), v_i, v_j \in V\}$ is the edge set, the number of nodes is given by $N = |V|$, where $|*|$ indicates the cardinal number of a set. The adjacency matrix of G is given by $A_{N\times N}$, in which $A_{ij} = 1$ if there is a link between node v_i and v_j, otherwise $A_{ij} = 0$. We use *DeepWalk* to learn the embedding vector of every node (Perozzi et al., 2014). Concretely speaking, *DeepWalk* conducts a uniform random walk to obtain node sequences as the input for a learning model, i.e., *SkipGram*. The embedding vectors of the nodes contain the structure information of the original network. For a node v_i, we use $\overrightarrow{V_i} = (v_{i1}, v_{i2}, \cdots, v_{id})$ to represent the embedding vector obtained from *DeepWalk*. Therefore, we can define the Euclidean distance between two arbitrary nodes v_i and v_j as $b_{ij} = \sqrt{\sum_{z=1}^{d}(v_{iz} - v_{jz})^2}$. Smaller b_{ij} indicates that v_i and v_j are more similar. The Euclidean distance matrix is denoted as $B_{N\times N}$, in which $B(i)$ is the Euclidean distance between node v_i and all the N nodes. Hence, we have $B(i,i) = 0$. We define $B_{\max} = \max_{i,j} B_{ij}$ and $B_{\min} = \min_{i,j} B_{ij} = 0$. We use $H_i = [H_{i1}, H_{i2}, \cdots, H_{iL}]$ to represent the Euclidean distance distribution of node v_i, in which H_{iz} is the probability that the Euclidean distance between a node and node v_i follows in the bin $\left[B_{\min} + (z - 1)\frac{B_{\max} - B_{\min}}{L}, B_{\min} + z\frac{B_{\max} - B_{\min}}{L}\right]$. L is a tunable parameter. It is worth noting that the distance used here is not limited to Euclidean distance. We test the robustness of our embedding method by using distance matrix generated by Manhattan distance and inner product between node embedding vectors. The performance of using these two distances for network comparison is further given in Figures S3–S4, which shows that different distance measures will not change the similarity trend of our network embedding-based comparison method.

We introduce Jensen-Shannon divergence to define the network dissimilarity based on the Euclidean distance distribution. The Euclidean distance distribution heterogeneity of a network, i.e., $JS(G)$, measures the heterogeneity of a network G in terms of the connectivity distances, and a network that possesses a high diversity of node distance patterns corresponds to a large $JS(G)$ value, which is defined as:

$$JS(G) = \frac{J(H_1, \ldots, H_N)}{\log(N+1)} \qquad \text{(Equation 1)}$$


A
B
C
G_1
v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8 v_9 v_{10} v_{11} v_{12}
G_2
v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8 v_9 v_{10} v_{11} v_{12}
DeepWalk
G_1
Node Embedding Vector
G_2
Node Euclidean Distance
Node Distance Distribution
D_{NE}
Network dissimilarity between G_1 and G_2
$JS(G_1) = 0.12$
$JS(G_2) = 0.18$
$D_{NE}(G_1, G_2) = 0.32$

	1	2	…	d - 1	d
v_1	0.034	0.089	…	-0.064	-0.065
v_2	0.032	0.077	…	-0.068	-0.067
…	…	…	…	…	…
v_{11}	0.038	0.072	…	-0.047	-0.054
v_{12}	0.037	0.066	…	-0.049	-0.052

	1	2	…	d - 1	d
v_1	-0.072	0.088	…	-0.086	-0.040
v_2	-0.019	0.057	…	-0.151	-0.082
…	…	…	…	…	…
v_{11}	-0.009	0.054	…	-0.047	-0.054
v_{12}	-0.023	0.034	…	-0.044	-0.044

	v_1	v_2	…	v_{11}	v_{12}
v_1	0.0	0.094	…	0.082	0.126
v_2	0.094	0.0	…	0.125	0.093
…	…	…	…	…	…
v_{11}	0.082	0.125	…	0.0	0.115
v_{12}	0.126	0.093	…	0.115	0.0

	v_1	v_2	…	v_{11}	v_{12}
v_1	0.0	0.410	…	0.485	1.143
v_2	0.410	0.0	…	0.768	1.210
…	…	…	…	…	…
v_{11}	0.485	0.768	…	0.0	1.205
v_{12}	1.143	1.210	…	1.205	0.0

	1	2	…	L - 1	L
H_1	0.09	0.09	…	0.00	0.00
H_2	0.18	0.00	…	0.00	0.00
…	…	…	…	…	…
H_{11}	0.09	0.09	…	0.00	0.00
H_{12}	0.18	0.00	…	0.09	0.09
μ_{G1}	0.18	0.05	…	0.11	0.07

	1	2	…	L - 1	L
H_1	0.18	0.18	…	0.00	0.09
H_2	0.18	0.00	…	0.00	0.09
…	…	…	…	…	…
H_{11}	0.09	0.09	…	0.00	0.09
H_{12}	0.00	0.00	…	0.00	0.09
μ_{G2}	0.16	0.09	…	0.02	0.13

Figure 1. Illustration of the network embedding-based comparison method

(A) Visualization of two networks G_1 and G_2, each with 12 nodes and 12 edges. It should be noted that our method is applicable to compare networks with different node and edge sizes.

(B–C) Example of how to compute network embedding-based dissimilarity, including the characterization of the node embedding, calculation of the node Euclidean distance, node distance distribution, average Euclidean distance distribution, and the dissimilarity between network G_1 and G_2, where we use $\omega = 0.5$.

where $J(H_1, \ldots, H_N) = \frac{1}{N}\sum_{i,j} H_i(j) log \frac{H_i(j)}{\mu_j}$ represents the Jensen-Shannon divergence of the node Euclidean distance distribution. The average Euclidean distance distribution for a network G is given by $\mu_G = \{\mu_1, \mu_2, \cdots, \mu_L\}$, in which $\mu_j = \frac{\sum_{i=1}^{N} H_i(j)}{N}$ $(j = 1 \cdots, L)$, i.e., μ_j is the average value of the j_{th} dimension of H and represents the average probability of nodes that have Euclidean distance falls in the bin $\left[B_{\min} + (j-1)\frac{B_{\max} - B_{\min}}{L}, B_{\min} + j\frac{B_{\max} - B_{\min}}{L}\right]$.

Given two networks $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, we denote μ_{G_1} and μ_{G_2} as the average Euclidean distance distributions of G_1 and G_2, respectively. The dissimilarity between G_1 and G_2 ($D_{NE}(G_1, G_2)$) is given by

$$D_{NE}(G_1, G_2) = \omega\sqrt{\frac{J(\mu_{G_1}, \mu_{G_2})}{\log 2}} + (1-\omega)\left|\sqrt{JS(G_1)} - \sqrt{JS(G_2)}\right|, \quad \text{(Equation 2)}$$

where $\omega \in [0, 1]$ is a tunable parameter that controls the extent of global and local differences while comparing two networks, and thus we have $D_{NE} \in [0, 1]$. The first term in Equation (2) compares the global dissimilarities between networks through calculating the average Euclidean distance distributions. The second term compares the local differences through evaluating the dissimilarity of Euclidean distance heterogeneity between two networks. Smaller value of $D_{NE}(G_1, G_2)$ indicates that G_1 and G_2 are more similar.

To obtain node embedding vector from *DeepWalk*, we set the parameters such as embedding dimension $d = 128$, number of walks per node $s = 10$, the length of each walk $l = 60$, and the context window size $w = 8$. In addition, we set $L = 10$ in the Euclidean distance distribution H_i $(i = 1, 2, \cdots, N)$. The influence of distribution length L and the parameters d, s, l, and w of *DeepWalk* on the performance of network comparison is further given in Figures S5–S16, which shows different settings of these parameters will not change the similarity trend between networks.

In Figure 1, we show the network dissimilarity comparison process of our method D_{NE}. In Figure 1A, we show two networks, i.e., G_1 and G_2, in which G_1 is a fully connected network and G_2 has one isolated

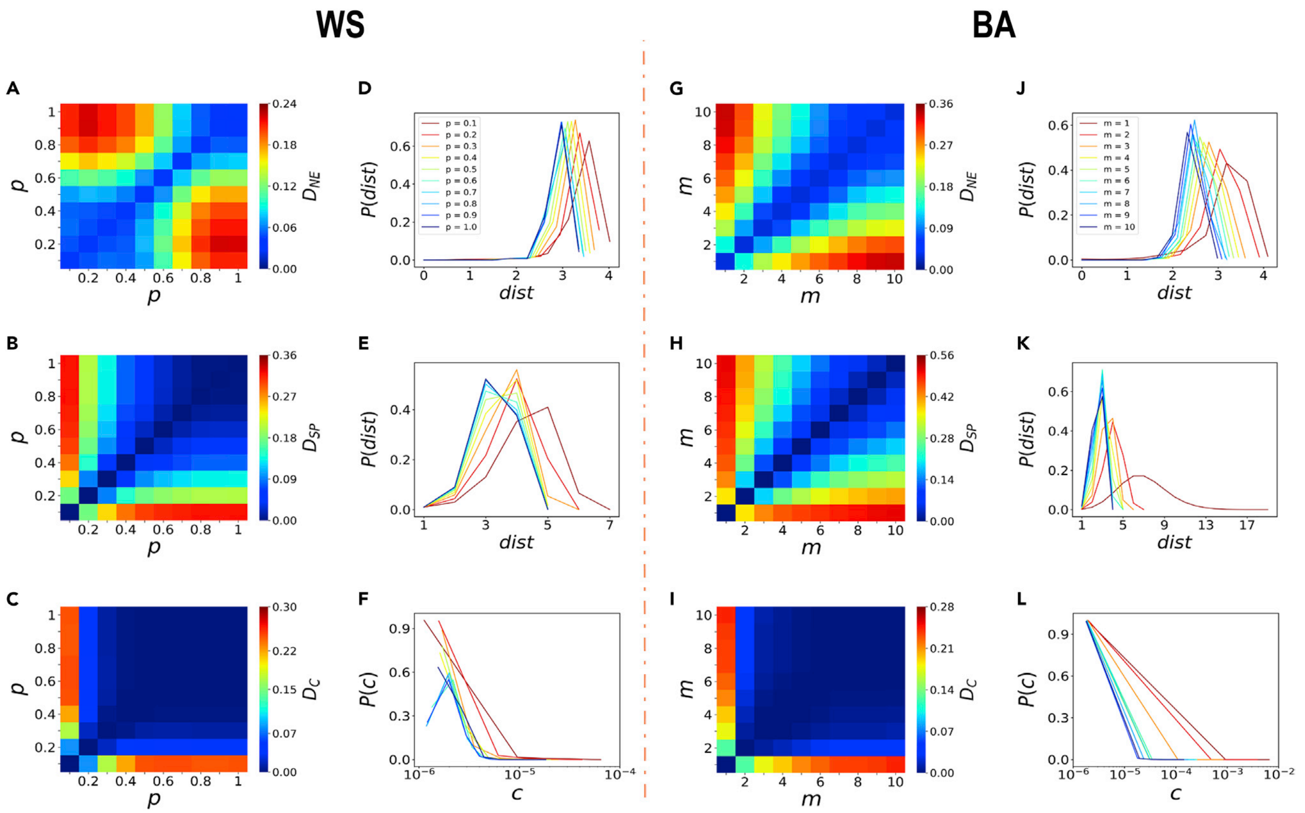


Figure 2. Performance of three comparison methods on synthetic networks

(A–C) Dissimilarity values D_{NE}, D_{SP}, and D_C of networks generated by WS model, respectively.

(D) The embedding-based average distance distributions of networks generated by WS model with different p based on D_{NE}.

(E) The average distance distributions of networks generated by WS model with different p based on D_{SP}.

(F) The node communicability distributions of networks generated by WS model with different p based on D_C.

(G–I) Dissimilarity values D_{NE}, D_{SP}, and D_C of networks generated by BA model under different m, respectively, in which $m \in \{1,2,3,4,5,6,7,8,9,10\}$.

(J) The embedding-based average distance distributions of networks generated by BA model with different m based on method D_{NE}.

(K) The average distance distribution of networks generated by BA model with different m based on method D_{SP}.

(L) The node communicability distribution of networks generated by BA model with different m based on method D_C. All the results are averaged over 100 realizations, where we use $\omega = 0.5$.

node. The detailed calculation process of the method is shown in Figures 1B and 1C, which include the calculation of node embedding vector, node Euclidean distance matrix, node distance distribution, and the dissimilarity between the two networks via Equations (1) and (2). The dissimilarity between G_1 and G_2 via D_{NE} is as high as 0.32.

Synthetic network comparison

To verify the ability of our method in quantifying the network dissimilarity, we perform the comparison on synthetic networks including networks generated by WS and BA models. In all the network models, we use the network size $N = 1000$. In WS model, we compare networks generated by different rewiring probability p, where the network average degree is 10. Figures 2A–2C show the dissimilarity values obtained by D_{NE}, D_{SP}, and D_C between networks generated by WS model with different p. Generally, we find that all the three kinds of dissimilarity values of the networks generated with similar p values are much smaller than those of the networks generated with dramatically different values of p. The proposed method D_{NE} can detect the network dissimilarity for all the p values (Figure 2A), while D_{SP} and D_C cannot identify the difference between networks for large values of p (Figures 2B and 2C). The definition of D_{NE}, D_{SP}, and D_C is based on the embedding-based distance distribution, the shortest path distance distribution, and the node communicability distribution, respectively. The embedding-based distance distributions are distinguishable across different p (Figure 2D). However, the distributions of shortest path distance and node

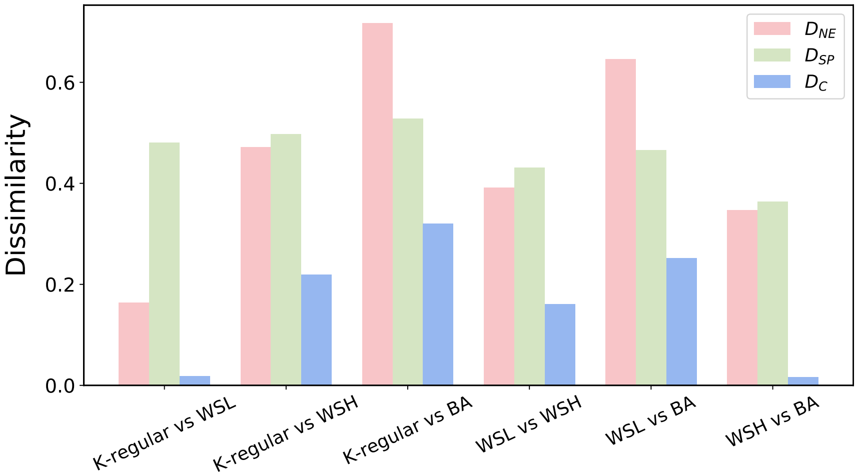


Figure 3. Comparison of four synthetic networks (K-regular, WSL, WSH, and BA)

We use three different methods, i.e., D_{NE}, D_{SP}, and D_C, respectively, in which D_{NE} is the method of network embedding, D_{SP} calculates the dissimilarity value based on the method using distance distribution, and D_C calculates the dissimilarity value based on the communication sequence entropy. We consider network size $N = 1000$, average node degree 10. All the results are averaged over 100 realizations with $\omega = 1.0$.

communicability are so narrow for large p values (Figures 2E and 2F), leading to no difference for the corresponding network comparison methods. Besides, the comparison of networks generated by WS model in the log-spaced of p is further given in Figures S1A–S1C, which reveals the same results as that of Figures 2A–2C. In BA model, we generate networks by changing the value of m, which is the number of edges per node added at each time step. Figures 2G–2I show the comparison of networks generated by BA model with $m \in [1, 10]$ via the three methods. Similarly as the WS model, D_{NE} shows the best performance. The reason that D_{SP} and D_C perform worse is given by the average shortest path distance distributions and the node communicability distributions when changing m in Figures 2K and 2L, respectively. In addition, we also compare the dissimilarities between the preferential attachment networks generated by different values of nonlinear preferential attachment kernel α in Figures S1D–S11F, which again shows our method outperforms the baselines.

We show how the dissimilarity between networks changes with the parameter ω in Figures S2A and S2B. In all the networks, we keep the average degree as 10. Each point in Figure S2A shows the dissimilarity between a network generated by WS model with size $N = 1000$ and $N = \{1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000\}$, respectively. We set rewiring probability as $p = 0.3$. Different curves show the dissimilarity when we use different ω. We find that a network generated by WS model with size $N = 1000$ is more similar to networks generated with close size, and different ω does not affect the similarity trend. However, large value of ω results in larger dissimilarity values between networks. In Figure S2B, we give the same analysis for BA model, which shows the similar results as those of WS model. We also compare the differences between the following networks: BA, WSL (it is obtained by rewiring 1% of edges in K-regular network), and WSH (it is obtained by rewiring 10% of edges in K-regular network) in Figure S2C. Figure S2C shows the change of the dissimilarity values with the increase of ω, and the results show that large value of ω gives large dissimilarity value. Furthermore, when $\omega = 0$, indicating that only local structural information is used (Equation (2)), the differences between the three pairs of synthetic networks are not effectively distinguished. On the contrary, when $\omega = 1$, the global information of the network can better distinguish the network differences. Therefore, we set $\omega = 1$ in the following analysis.

To compare with different dissimilarity methods, we also show the dissimilarity between four synthetic networks with the same node size $N = 1000$, edge size $|E| = 5000$ and average node degree 10. The four networks include K-regular, WSL, WSH, and BA model. From the generation model, we know that the descending order of similarity value between K-regular and the other three networks is as follows: WSL, WSH, and BA. Figure 3 gives the dissimilarity between the four networks with three methods, i.e., D_{NE}, D_{SP}, and D_C. Figure 3 implies that dissimilarities between the four networks obtained by the three network comparison methods are consistent with the rules of network generation models. However, the dissimilarity values

Table 1. Basic properties of real networks

Networks	N	$\|E\|$	Ad	Avl	Ld	C	dia
Pgp	10,680	24,316	4.55	7.49	0.0004	0.266	24
Yeast	1,870	2,203	2.44	6.81	0.0013	0.067	19
Contiguous	49	107	4.37	4.16	0.0910	0.497	11
Infectious	410	2,765	13.49	3.63	0.0330	0.456	9
Rovira	1,133	5,451	9.62	3.61	0.0085	0.220	8
Petsterc	2,426	16,631	13.71	3.59	0.0057	0.538	10
Petster	1,858	12,534	13.49	3.45	0.0073	0.141	14
Irvine	1,899	59,835	14.57	3.06	0.0079	0.109	8
Metabolic	453	2,025	8.94	2.68	0.0198	0.646	7
Jazz	198	2742	27.69	2.24	0.1406	0.617	6
Chesapeake	39	170	8.72	1.83	0.2294	0.450	3
Windsurfers	43	336	15.63	1.69	0.3721	0.653	3

N, $|E|$, Ad, Avl, Ld, C, and dia represent the number of nodes, the number of edges, average degree, average shortest path length, link density, average clustering coefficient, and network diameter, respectively.

(D_{SP}) between K-regular and WSL, K-regular and WSH are almost the same and the dissimilarity values (D_{SP}) between the four synthetic networks are very close, indicating that the method D_{SP} cannot effectively discriminate the differences between these synthetic networks.

Real networks comparison

We validate the effectiveness of our network embedding-based comparison method upon real networks from different domains. Table 1 gives the basic properties of the real networks, including the number of nodes (N), the number of edges ($|E|$), average degree (Ad), average path length (Avl), link density (Ld), clustering coefficient (C), and diameter (dia). The 12 real networks range from the protein-protein interaction (Yeast) and metabolic network (Metabolic) in biology, to the human contact (Infectious, Windsurfers), and to the social communication networks (Pgp, Rovira, Petster, Petsterc, and Irvine).

Firstly, we show the difference between a real network and its corresponding null models in Figure 4A. For a network G, we consider three kinds of null models (k-order null models, including k = 1.0, 2.0, and 2.5) (Orsini et al., 2015), which is defined as $Dk1.0$, $Dk2.0$, and $Dk2.5$, respectively. Specifically, different values of k indicate the preservation of network topology to different degrees. k = 1.0 indicates that the generated network retains the degree sequence. When k = 2.0, the degree sequence and degree correlation properties are invariant during the rewiring process. k = 2.5 preserves the clustering spectrum property of the original network. The dissimilarity values are averaged over 100 repeated independent runs. With the increase of k, the dissimilarity between a real network and its randomized networks tends to be smaller across different networks (each row in Figure 4A). The pattern of the network dissimilarity values is consistent with the randomization process, where larger k indicates that the randomized networks share more properties with the original network, leading to the more similarity to the original network.

We also compare the real networks with the networks after certain perturbation. The perturbation is performed as follows: for a given network, we randomly add (or delete) a certain fraction $f \in [0,1]$ of edges, and then compare the dissimilarity between the original network and the perturbed network. Positive f represents addition process, and negative f represents deletion process. Figure 4B shows the dissimilarity between Petsterc network and the perturbed networks after random addition or deletion of edges. It implies that the more we perturb the network, the more dissimilar it is to the original network. We show the similar trend of the other networks in Figure S6. The results indicate that our comparison method can distinguish the differences between a real network and the networks generated after certain perturbation.

Analysis on the hybrid method

Figure 2 shows D_{NE} is an effective way to distinguish networks and shortest path distance-based method (D_{SP}) can partly tell the difference between different synthetic networks. We further hybridize these two

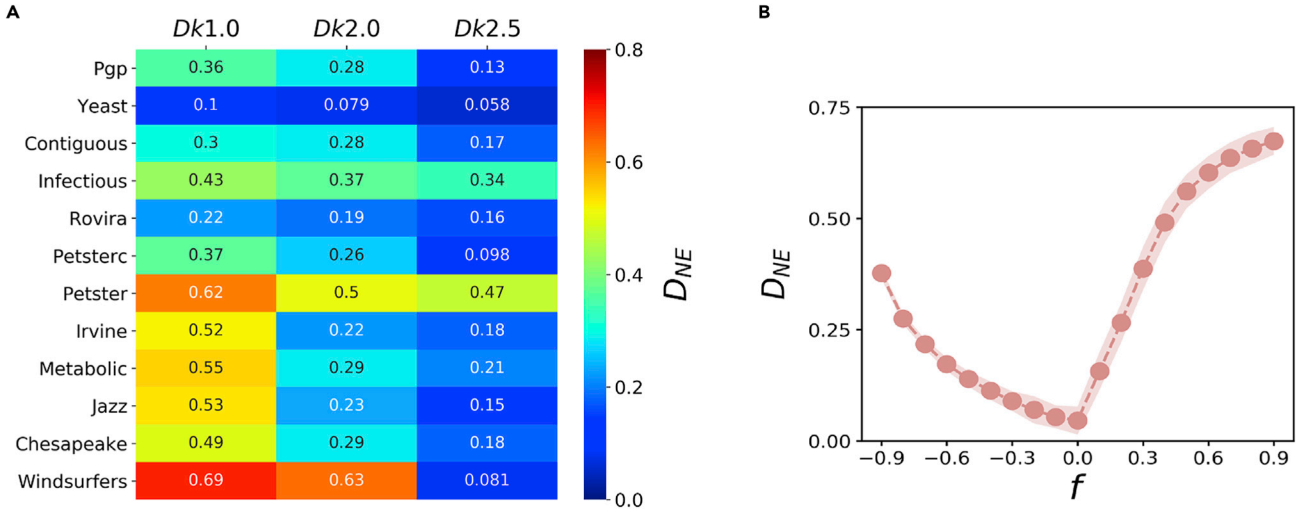


Figure 4. Dissimilarity between real networks

(A) Comparison between real networks and their null models. We consider the *Dk* models with different *k*-values (1.0, 2.0, and 2.5).

(B) Dissimilarity between Petsterc network and the networks generated after certain perturbations, where negative value of *f* corresponds to the random deletion of edges with the given ratios, and vice versa. Each point in the figure is averaged over 100 times. The shaded error area shows the standard deviation of 100 times.

distance distributions to explore the performance of the hybrid method on network comparison. To recap, we use $P_{N\times N}$ and $H_{N\times L}$ to represent the shortest path distance distribution and the distance distribution based on network embedding, respectively. As the dimension of p and H is different, we expand short dimension matrix with zero values. That is to say, if $N < L$, we expand P_i to $L\times 1$ dimensions, i.e., $P_i = (P_{i1}, P_{i2}, \cdots, P_{iN}, 0, \cdots, 0)$. And if $N > L$, we expand H_i to $N\times 1$ dimensions, i.e., $H_i = (H_{i1}, H_{i2}, \cdots, H_{iL}, 0, \cdots, 0)$. For each node v_i, the hybrid distance distribution M_i is defined as the normalization of $\lambda P_i + (1 - \lambda)H_i$. Thus, we define the hybrid distance distribution M as

$$M = \lambda P + (1 - \lambda)H, \qquad \text{(Equation 3)}$$

where λ is a tunable parameter. We use M to replace H in Equation (2), and obtain the hybrid network comparison method, which is denoted as D_M.

We test the performance of D_M on the comparison of a network and its null models (i.e., *Dk*1.0, *Dk*2.0, and *Dk*2.5) in Figure 5. We use $D_M(Dk1.0)$, $D_M(Dk2.0)$, and $D_M(Dk2.5)$ to represent the dissimilarity between the original network and its null models, respectively. The pattern of the dissimilarity between a real network and its null models (i.e., $D_M(Dk1.0) > D_M(Dk2.0) > D_M(Dk2.5)$) when $\lambda < 1$ is consistent with the order of the null models. However, D_M cannot tell the difference between the network and its null models very well, i.e., $D_M(Dk2.0)$ and $D_M(Dk2.5)$ share the same value in Figures 5H, 5I, and 5K when $\lambda = 1$. In fact, $\lambda = 1$ indicates that the hybrid distance distribution degrades into only considering the shortest path distance distribution (Equation (3)), leading to $D_M \approx D_{SP}$ for $\lambda = 1$. The network basic features show that the average shortest path length of Irvine (Figure 5H), Metabolic (Figure 5I), Chesapeake (Figure 5K), and Windsurfers (Figure 5L) are significantly smaller than the other networks, which cannot be well compared according to D_M for $\lambda = 1$. It indicates that D_{SP} cannot well tell the difference of the real network with small average shortest path length, which is consistent with the findings in the synthetic networks (Figures 2B and 2E). And for $\lambda = 0$, which means the hybrid method degrades into D_{NE}, shows better discriminative performance on network comparison across networks with different average shortest path length, which implies the robustness of D_{NE} upon different network structure.

Comparison between real networks

The dissimilarity between the 12 real networks is given in Figure 6. We show D_{NE} between network pairs in Figure 6A; we find that networks that have the similar value of average shortest path length tend to be similar. It implies that D_{NE} considers the path properties of a network when comparing networks. The implication is further amplified by the high Pearson correlation coefficient ($r = 0.50, p = 2.7 \times 10^{-5}$) between

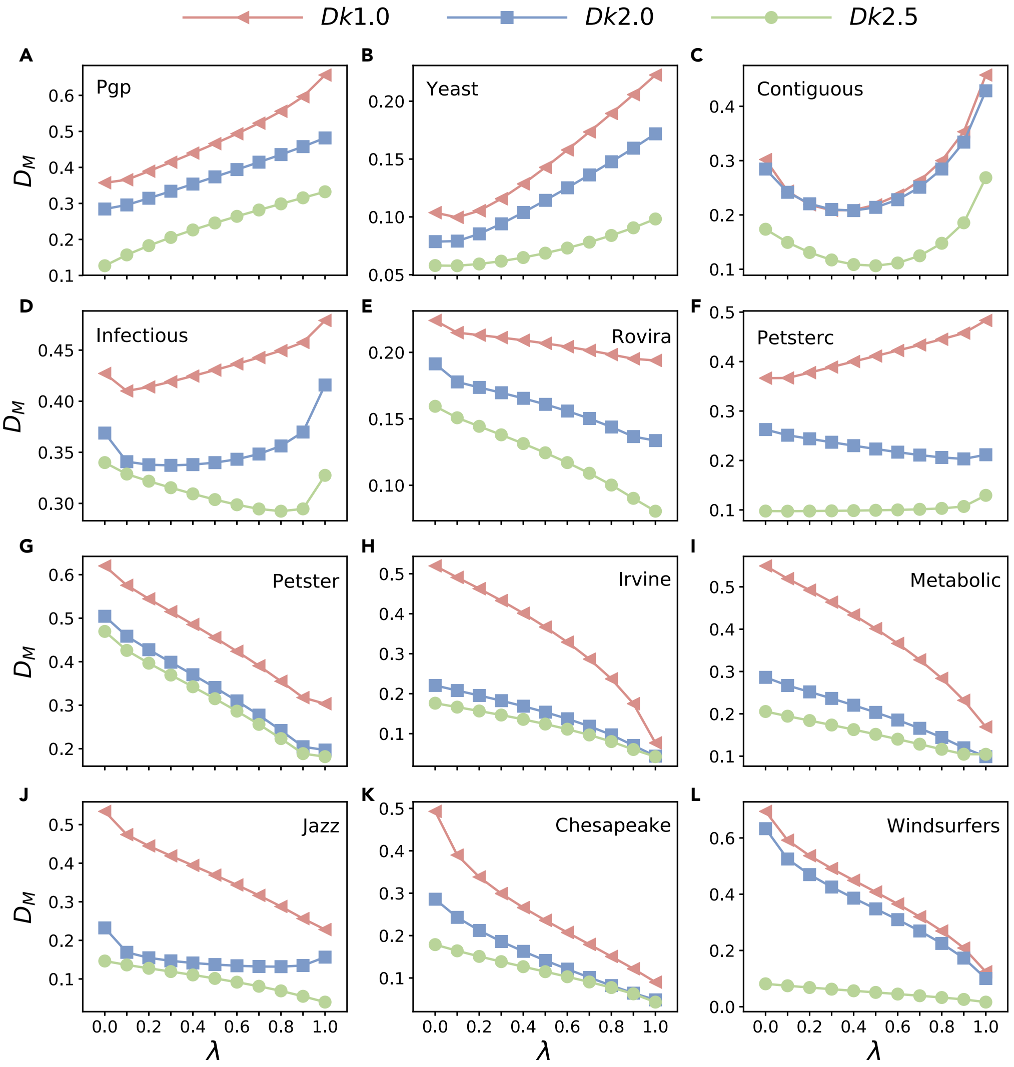


Figure 5. The dissimilarity between a network and its null models characterized by the hybrid method

When parameter λ = 0, the hybrid method degenerates to D_{NE}; when λ = 1, the real shortest path distance distribution of the network is used to characterize the dissimilarity. The red line in each figure describes the dissimilarities between the real network and the DK1.0 changing with the parameter λ. The blue line in each figure describes the dissimilarities between the real network and the DK2.0 changing with the parameter λ. The green line in each figure describes the dissimilarities between the real network and the DK2.5 changing with the parameter λ.

D_{SP} and D_{NE} given in Figure 6B, where the values of D_{SP} and D_{NE} are computed between the 12 real networks. Given two networks G_1 and G_2, we define the average shortest path length difference and the link density difference between them as $\Delta Avl = |Avl_1 - Avl_2|$ and $\Delta Ld = |Ld_1 - Ld_2|$, respectively. In Figure 6C, we show the Pearson correlation between D_{NE} and ΔAvl, which is as high as $r = 0.43$ ($p = 3.4 \times 10^{-4}$). It further explains the results of Figure 6A, i.e., networks with similar average shortest path length tend to be similar. Meanwhile, the high Pearson correlation coefficient (Figure 6D, $r = 0.42$, $p = 4.9 \times 10^{-4}$) is also found between D_{NE} and ΔLd. In conclusion, the network embedding-based comparison method can capture network properties such as average shortest path length and link density.

Modularity reflects the strength of division of a network into communities (Newman, 2006), i.e., a network with a high modularity indicates that nodes are densely connected within the communities and sparsely connected across different communities. Thus, we explore the relationship between modularity and network structural difference. We define the community segmentation with the maximal network modularity as Q, which corresponds to the optimal division of a network (Newman and Girvan, 2004). Given

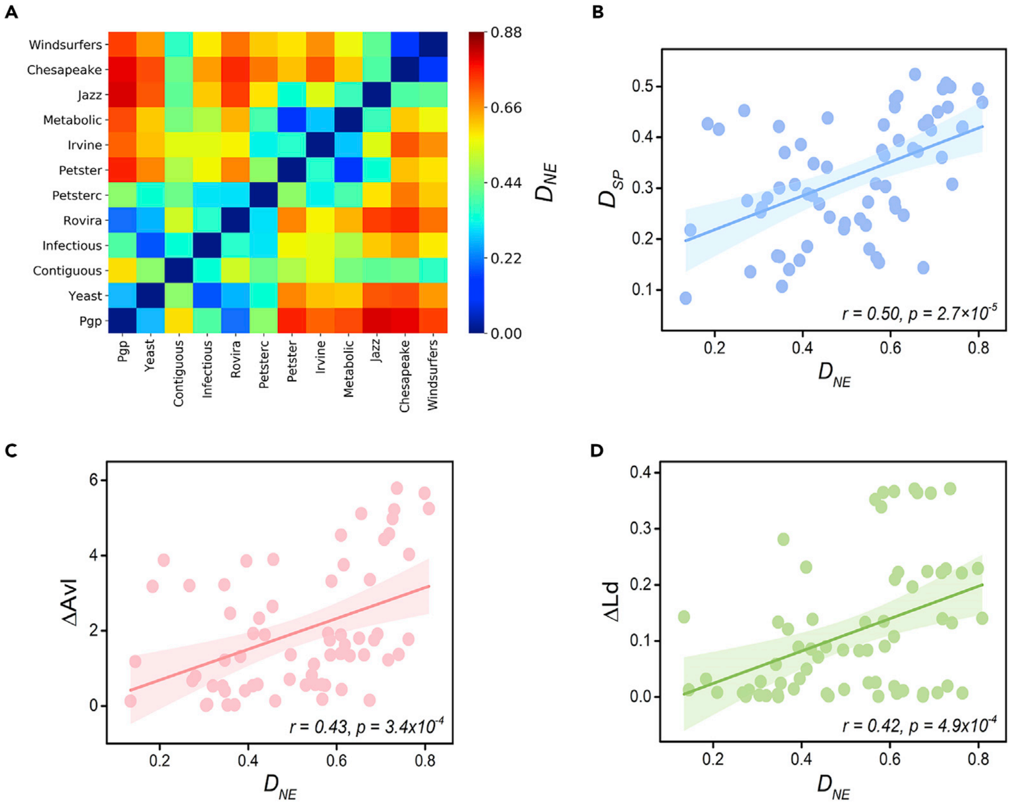


Figure 6. Correlation analysis in real networks

(A) Comparison between real networks, in which networks are sorted in descending order based on average shortest path length.

(B) Correlation between network comparison methods D_{SP} and D_{NE}.

(C) Correlation between network dissimilarities D_{NE} and average shortest path length differences ΔAvl on 12 real networks.

(D) Correlation between network dissimilarities D_{NE} and link density differences ΔLd on 12 real networks. Where r value shows the Pearson correlation, p value shows the assumption probability and the shaded error area shows the confidence interval.

two networks G_1 and G_2, we define the modularity difference between them as $\Delta Q = |Q_1 - Q_2|$. Figure 7 shows the correlation between ΔQ and dissimilarity value D_{NE} on 12 real networks. The result shows that the similar networks tend to have small value of ΔQ and vice versa. It further emphasizes the good performance of our network embedding-based comparison method.

DISCUSSION

In this paper, we propose a network embedding-based comparison method D_{NE}, which is based on node distance distribution and Jensen-Shannon divergence. Specifically, we firstly obtain the embedding vector for each node through *DeepWalk* and calculate the Euclidean distance between each of the node pairs. We measure the distance distribution heterogeneity of a network via defining the Jensen-Shannon divergence of the node distance distributions. The dissimilarity between two networks is further defined by the combination of the difference of the average distance distribution of the networks and the network Euclidean distance distribution heterogeneity. We compare the proposed method D_{NE} with two state-of-the-art methods, i.e., network dissimilarity based on shortest path distance distribution (D_{SP}) and network dissimilarity based on communicability sequence (D_C), on various synthetic and real networks. Furthermore, we find that D_{NE} shows better performance in quantifying network difference in almost all the networks. In addition, we find that D_{NE} is also linearly correlated with D_{SP} (Pearson correlation coefficient $r = 0.5$), and thus can capture network properties such as average shortest path length and link density. Moreover, it shows that real networks that are similar to each other tend to have small difference in modularity.

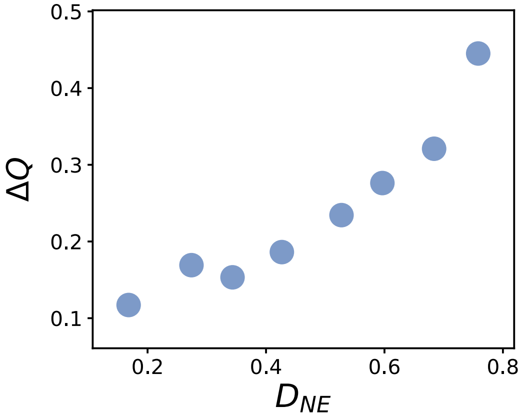


Figure 7. Correlation between network dissimilarities D_{NE} and modularity differences ΔQ on 12 real networks

We confined ourselves to *DeepWalk* to embed networks, which is a simple and efficient network embedding method. According to previous work, more advanced embedding methods, such as *Node2Vec* (Grover and Leskovec, 2016) and graph neural network (Zhang and Chen, 2018), can better capture the topology of the network, generating better performance in tasks such as link prediction, clustering, and classification. Therefore, these embedding methods could be promising in quantifying network dissimilarity. We deem that our methods can also be generalized to other network types, such as multilayer networks (Kivelä et al., 2014), temporal networks (Holme, 2015), signed networks (Wang et al., 2017), and hypergraphs (Feng et al., 2019).

Limitations of the study

The distance distribution used in our comparison method is based on a random walk embedding algorithm, i.e., *DeepWalk*, which is a black box model. Therefore, it is hard to theoretically deduce the specific properties that can be captured by the comparison method.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - ○ Lead contact
  - ○ Materials availability
  - ○ Data and code availability
- EXPERIMENTAL MODEL AND SUBJECT DETAILS
- METHOD DETAILS
  - ○ Datasets
  - ○ Baselines
- QUANTIFICATION AND STATISTICAL ANALYSIS
- ADDITIONAL RESOURCES

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2022.104446.

ACKNOWLEDGMENTS

This work was supported by Natural Science Foundation of Zhejiang Province (Grant Nos. LQ22F030008 and LR18A050001), the National Natural Science Foundation of China (Grant Nos. 92146001, 61873080 and 61673151), the Major Project of the National Social Science Fund of China (Grant No. 19ZDA324), and the Scientific Research Foundation for Scholars of HZNU (2021QDL030).

AUTHOR CONTRIBUTIONS

All authors planed the study. All authors performed the experiments and prepared the figures. All authors analyzed the results and wrote the manuscript. All authors read and approved the final manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: November 26, 2021
Revised: May 1, 2022
Accepted: May 17, 2022
Published: June 17, 2022

REFERENCES

Ali, W., Rito, T., Reinert, G., Sun, F., and Deane, C.M. (2014). Alignment-free protein interaction network comparison. Bioinformatics *30*, i430–i437. https://doi.org/10.1093/bioinformatics/btu447.

Babai, L. (2016). Graph isomorphism in quasipolynomial time. In Proc. 48th Ann. ACM Symp. Theory Comput (ACM Press), pp. 684–697.

Bagrow, J.P., and Bollt, E.M. (2019). An information-theoretic, all-scales approach to comparing networks. Appl. Math. Comput. *4*, 45. https://doi.org/10.1007/s41109-019-0156-x.

Barabási, A.-L., and Albert, R. (1999). Emergence of scaling in random networks. Science *286*, 509–512. https://doi.org/10.1126/science.286.5439.509.

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D.-U. (2006). Complex networks: structure and dynamics. Phys. Rep. *424*, 175–308. https://doi.org/10.1016/j.physrep.2005.10.009.

Bu, Z., Wang, Y., Li, H.-J., Jiang, J., Wu, Z., and Cao, J. (2019). Link prediction in temporal networks: integrating survival analysis and game theory. Inf. Sci. *498*, 41–61. https://doi.org/10.1016/j.ins.2019.05.050.

Bullmore, E., and Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and functional systems. Nat. Rev. Neurosci. *10*, 186–198. https://doi.org/10.1038/nrn2575.

Chen, D., Shi, D.-D., Qin, M., Xu, S.-M., and Pan, G.-J. (2018). Complex network comparison based on communicability sequence entropy. Phys. Rev. E *98*, 012319. https://doi.org/10.1103/physreve.98.012319.

Costa, L.d.F., Rodrigues, F.A., Travieso, G., and Villas Boas, P.R. (2007). Characterization of complex networks: a survey of measurements. Adv. Phys. *56*, 167–242. https://doi.org/10.1080/00018730601170527.

De Domenico, M., and Biamonte, J. (2016). Spectral entropies as information-theoretic tools for complex network comparison. Phys. Rev. X *6*, 041062. https://doi.org/10.1103/physrevx.6.041062.

De Domenico, M., Nicosia, V., Arenas, A., and Latora, V. (2015). Structural reducibility of multilayer networks. Nat. Commun. *6*, 6864–6869. https://doi.org/10.1038/ncomms7864.

Feng, Y., You, H., Zhang, Z., Ji, R., and Gao, Y. (2019). Hypergraph neural networks. Proc. AAAI Conf. Artif. Intell. *33*, 3558–3565. https://doi.org/10.1609/aaai.v33i01.33013558.

Fortunato, S. (2010). Community detection in graphs. Phys. Rep. *486*, 75–174. https://doi.org/10.1016/j.physrep.2009.11.002.

Gärtner, T., Flach, P., and Wrobel, S. (2003). On graph kernels: hardness results and efficient alternatives. In Learn. Theor. Kernel Mach. (Springer), pp. 129–143. https://doi.org/10.1007/978-3-540-45167-9_11.

Goyal, P., and Ferrara, E. (2018). Graph embedding techniques, applications, and performance: a survey. Knowl.-Based Syst. *151*, 78–94. https://doi.org/10.1016/j.knosys.2018.03.022.

Grohe, M., and Schweitzer, P. (2020). The graph isomorphism problem. Commun. ACM *63*, 128–134. https://doi.org/10.1145/3372123.

Grover, A., and Leskovec, J. (2016). node2vec: scalable feature learning for networks. In Proc. 22nd ACM SIGKDD Intl. Conf. Knowl. Disc. Data Min. (ACM Press), pp. 855–864. https://doi.org/10.1145/2939672.2939754.

Hartle, H., Klein, B., McCabe, S., Daniels, A., St-Onge, G., Murphy, C., and Hébert-Dufresne, L. (2020). Network comparison and the within-ensemble graph distance. Proc. R. Soc. A. *476*, 20190744. https://doi.org/10.1098/rspa.2019.0744.

Holme, P. (2015). Modern temporal network theory: a colloquium. Eur. Phys. J. B *88*, 234–330. https://doi.org/10.1140/epjb/e2015-60657-4.

Holme, P., and Saramäki, J. (2012). Temporal networks. Phys. Rep. *519*, 97–125. https://doi.org/10.1016/j.physrep.2012.03.001.

Jin, D., You, X., Li, W., He, D., Cui, P., Fogelman-Soulié, F., and Chakraborty, T. (2019). Incorporating network embedding into Markov random field for better community detection. Proc. AAAI Conf. Artif. Intell. *33*, 160–167. https://doi.org/10.1609/aaai.v33i01.3301160.

Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J.P., Moreno, Y., and Porter, M.A. (2014). Multilayer networks. J. Comput. Sci. *2*, 203–271. https://doi.org/10.1093/comnet/cnu016.

Li, J., Zhu, J., and Zhang, B. (2016). Discriminative deep random walk for network classification. In Proc. 54th Ann. Meeting Assoc. Comput. Linguist. (ACL Press), pp. 1004–1013. https://doi.org/10.18653/v1/p16-1095.

Liu, C., Ma, Y., Zhao, J., Nussinov, R., Zhang, Y.-C., Cheng, F., and Zhang, Z.-K. (2020). Computational network biology: data, models, and applications. Phys. Rep. *846*, 1–66. https://doi.org/10.1016/j.physrep.2019.12.004.

Lu, S., Kang, J., Gong, W., and Towsley, D. (2014). Complex network comparison using random walks. In Proc. 23th Intl. Conf. World Wide Web (ACM Press), pp. 727–730.

Martínez, J.H., and Chavez, M. (2019). Comparing complex networks: in defence of the simple. New J. Phys. *21*, 013033. https://doi.org/10.1088/1367-2630/ab0065.

Newman, M.E.J. (2006). Modularity and community structure in networks. Proc. Natl. Acd. Sci. USA *103*, 8577–8582. https://doi.org/10.1073/pnas.0601602103.

Newman, M.E.J., and Girvan, M. (2004). Finding and evaluating community structure in networks. Phys. Rev. E. *69*, 026113. https://doi.org/10.1103/physreve.69.026113.

Orsini, C., Dankulov, M.M., Colomer-de-Simón, P., Jamakovic, A., Mahadevan, P., Vahdat, A., Bassler, K.E., Toroczkai, Z., Boguná, M., Caldarelli, G., et al. (2015). Quantifying randomness in real networks. Nat. Commun. *6*, 8627–8710. https://doi.org/10.1038/ncomms9627.

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk: online learning of social representations. In Proc. 20th ACM SIGKDD Intl. Conf. Knowl. Disc. Data Min. (ACM Press), pp. 701–710. https://doi.org/10.1145/2623330.2623732.

Pio, G., Ceci, M., Prisciandaro, F., and Malerba, D. (2020). Exploiting causality in gene network reconstruction based on graph embedding. Mach. Learn. *109*, 1231–1279. https://doi.org/10.1007/s10994-019-05861-8.

Saxena, R., Kaur, S., and Bhatnagar, V. (2019). Identifying similar networks using structural hierarchy. Physica A *536*, 121029. https://doi.org/10.1016/j.physa.2019.04.265.

Schieber, T.A., Carpi, L., Díaz-Guilera, A., Pardalos, P.M., Masoller, C., and Ravetti, M.G. (2017). Quantification of network structural dissimilarities. Nat. Commun. *8*, 13928–14010. https://doi.org/10.1038/ncomms13928.

Strogatz, S.H. (2001). Exploring complex networks. Nature *410*, 268–276. https://doi.org/10.1038/35065725.

Tsitsulin, A., Mottin, D., Karras, P., Bronstein, A., and Müller, E. (2018). Netlsd: hearing the shape of a graph. In Proc. 24th ACM SIGKDD Intl. Conf. Knowl. Disc. Data Min. (ACM Press), pp. 2347–2356.

Wang, S., Tang, J., Aggarwal, C., Chang, Y., and Liu, H. (2017). Signed network embedding in social media. In Proc. 2017 SIAM Intl. Conf. Data Min. (SIAM Press), pp. 327–335.

Watts, D.J., and Strogatz, S.H. (1998). Collective dynamics of small-world networks. Nature *393*, 440–442. https://doi.org/10.1038/30918.

Xu, S., Zhang, C., Wang, P., and Zhang, J. (2020). Variational bayesian weighted complex network reconstruction. Inf. Sci. *521*, 291–306. https://doi.org/10.1016/j.ins.2020.02.050.

Zemlyachenko, V.N., Korneenko, N.M., and Tyshkevich, R.I. (1985). Graph isomorphism problem. J. Sov. Math. *29*, 1426–1481. https://doi.org/10.1007/bf02104746.

Zhan, X.-X., Liu, C., Zhou, G., Zhang, Z.-K., Sun, G.-Q., Zhu, J.J., and Jin, Z. (2018). Coupling dynamics of epidemic spreading and information diffusion on complex networks. Appl. Math. Comput. *332*, 437–448. https://doi.org/10.1016/j.amc.2018.03.050.

Zhang, M., and Chen, Y. (2018). Link prediction based on graph neural networks. In Proc. 32nd Intl. Conf. Neural Inform. Proc. Syst. (NIPS Press), pp. 5171–5181.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Network data	This paper	https://doi.org/10.5281/zenodo.6526610
Code	This paper	https://doi.org/10.5281/zenodo.6526610
Software and algorithms		
Python version 3.6	Python Software Foundation	https://www.python.org
OriginPro 9.1	Data Analysis and Graphing Software	https://www.originlab.com/

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contacts, Xiu-Xiu Zhan (zhanxiuxiu@hznu.edu.cn), or Zi-Ke Zhang (zkz@zju.edu.cn).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- This paper analyzes existing, publicly available data. The DOI is listed in the key resources table.
- All original code has been deposited on GitHub through Zenodo. The DOI is listed in the key resources table.
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

EXPERIMENTAL MODEL AND SUBJECT DETAILS

This work does not use experimental models typical in the life sciences.

METHOD DETAILS

Datasets

We consider 12 kinds of real networks, the description of each network is as follows:

- Pgp is a interaction network of users of the Pretty Good Privacy (PGP) algorithm, which only contains the giant connected component.
- Yeast and Metabolic are the biological networks, in which Yeast is the protein interaction network and Metabolic is a metabolic network of *Caenorhabditis elegans*.
- Contiguous is a regional border network in the United States excepted isolated states Alaska and Hawaii.
- Rovira is an e-mail communication network at the University Rovira i Virgili in Tarragona in the south of Catalonia in Spain.
- Petsterc and Petster contain friendships and family links between users of the website, in which Petster is the giant connected component.
- Irvine is a messaging network between the users of an online community of students from the University of California, Irvine.

- Jazz is a collaboration network between Jazz musicians.
- Chesapeake is a mesohaline trophic network of Chesapeake Bay, an estuary in the United States of America.
- Windsurfers contains interpersonal contacts between windsurfers in southern California during the fall of 1986.

Baselines

Network dissimilarity based on shortest path distance distribution

Suppose the shortest path distance distribution of node v_i is denoted by $P_i = \{p_i(j)\}$, in which $p_i(j)$ is defined as the fraction of nodes at distance j from node v_i. Network node dispersion NND measures the network connectivity heterogeneity in terms of shortest path distance and is defined by the following equation:

$$NND(G) = \frac{J(P_1, \ldots, P_N)}{\log(dia+1)}, \qquad \text{(Equation 4)}$$

where dia represents the network diameter and $J(P_1, \ldots, P_N)$ is the Jensen-Shannon divergence of the node distance distribution. The dissimilarity measure D_{SP} is based on three distance-based probability distribution function vectors and is defined as follows:

$$D_{SP}(G_1, G_2) = \omega_1 \sqrt{\frac{J(\mu_{G_1}, \mu_{G_2})}{\log 2}} + \omega_2 \left| \sqrt{NND(G_1)} - \sqrt{NND(G_2)} \right| + \frac{\omega_3}{2} \left(\sqrt{\frac{J(P_{\alpha G_1}, P_{\alpha G_2})}{\log 2}} + \sqrt{\frac{J\left(P_{\alpha G_1^c}, P_{\alpha G_2^c}\right)}{\log 2}} \right), \qquad \text{(Equation 5)}$$

where ω_1, ω_2, ω_3, and α are tunable parameters, in which $\omega_1 + \omega_2 + \omega_3 = 1$. The first term in Equations (4) and (5) indicates the dissimilarity characterized by the averaged shortest path distance distributions, i.e., μ_{G_1} and μ_{G_2}. The second term characterizes the difference of network node dispersion. The last term is the difference of the α-centrality distributions, in which G^c is the complementary graph of G. We set $\omega_1 = \omega_2 = 0.45$ and $\omega_3 = 0.1$, which are the default settings used in (Schieber et al., 2017).

Network dissimilarity based on communicability sequence

The communicability matrix C measures the communicability between nodes and is defined as follows:

$$C = e^A = \sum_{z=0}^{\infty} \frac{1}{z!} A^z = \left\{ \begin{matrix} c_{11} & c_{12} & \cdots & c_{1N} \\ c_{21} & c_{22} & \cdots & c_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N1} & c_{N2} & \cdots & c_{NN} \end{matrix} \right\}, \qquad \text{(Equation 6)}$$

where c_{ij} unveils the communicability between node v_i and v_j. Let $P = \{P_1, P_2, \cdots, P_M\}$ be the normalized communicability sequence, in which $P_z = \frac{c_{ij}}{\sum_{i=1}^{N} \sum_{j=i}^{N} c_{ij}}$ $(1 \le z \le M,\ 1 \le i \le j \le N$ and $M = \frac{N(N+1)}{2})$. The Jensen-Shannon entropy of the sequence is expressed as follows:

$$S(P) = -\sum_{i=1}^{M} P_i \log_2 P_i \qquad \text{(Equation 7)}$$

Given two networks G_1 and G_2, normalized communicability sequences are given by P^{G_1} and P^{G_2}, respectively. We sort the values in P^{G_1} (P^{G_2}) in an ascending order and obtain new communicability sequences as $\tilde{P}^{G_1}$ $(\tilde{P}^{G_2})$. Therefore, the communicability based dissimilarity is defined as $D_C(G_1, G_2)$:

$$D_C(G_1, G_2) = S\left(\frac{\tilde{P}^{G_1} + \tilde{P}^{G_2}}{2} \right) - \frac{1}{2} \left[S\left(\tilde{P}^{G_1}\right) + S\left(\tilde{P}^{G_2}\right) \right] \qquad \text{(Equation 8)}$$

QUANTIFICATION AND STATISTICAL ANALYSIS

We give the average dissimilarities between a pair of networks for 100 runs and give the standard deviation of the dissimilarities between the original real networks and the networks generated after certain perturbations. The confidence interval, Pearson correlation coefficient and p value in Figure 6 are calculated by Origin.

ADDITIONAL RESOURCES

This work does not include any additional resources.