

## Identifying networks with common organizational principles

ANATOL E. WEGNER\*,

*University College London, Department of Statistical Science, Gower Street, London WC1E 6BT, UK*

*University of Oxford, Department of Statistics, 24-29 St. Giles', Oxford, OX1 3LB, UK*

\*Corresponding author: a.wegner@ucl.ac.uk

LUIS OSPINA-FORERO

*University of Oxford, Department of Statistics, 24-29 St. Giles', Oxford, OX1 3LB, UK*

*luis.ospinaforero@linacre.ox.ac.uk*

ROBERT E. GAUNT

*The University of Manchester, School of Mathematics, Manchester M13 9PL, UK*

*University of Oxford, Department of Statistics, 24-29 St. Giles', Oxford, OX1 3LB, UK*

*robert.gaunt@manchester.ac.uk*

CHARLOTTE M. DEANE

*University of Oxford, Department of Statistics, 24-29 St. Giles', Oxford, OX1 3LB, UK*

*deane@stats.ox.ac.uk*

AND

GESINE REINERT

*University of Oxford, Department of Statistics, 24-29 St. Giles', Oxford, OX1 3LB, UK*

*reinert@stats.ox.ac.uk*

Many complex systems can be represented as networks, and the problem of network comparison is becoming increasingly relevant. There are many techniques for network comparison, from simply comparing network summary statistics to sophisticated but computationally costly alignment-based approaches. Yet it remains challenging to accurately cluster networks that are of a different size and density, but hypothesized to be structurally similar. In this paper, we address this problem by introducing a new network comparison methodology that is aimed at identifying common organizational principles in networks. The methodology is simple, intuitive and applicable in a wide variety of settings ranging from the functional classification of proteins to tracking the evolution of a world trade network.

*Keywords:* networks | network comparison | machine learning | earth mover's distance | network topology

### 1. Introduction

Many complex systems can be represented as networks, including friendships, the World Wide Web, global trade flows and protein-protein interactions [Newman, 2010]. The study of networks has been a very active area of research in recent years, and in particular, network comparison has become increasingly relevant e.g. [Wilson and Zhu, 2008, Neyshabur et al., 2013, Ali et al., 2014, Yaveroglu et al., 2014]. Network comparison itself has many wide-ranging applications, for example, comparing protein-

protein interaction networks could lead to increased understanding of underlying biological processes [Singh et al., 2008, Ali et al., 2014]<sup>AEW: [Reference added.]</sup>. Network comparison can also be used to study the evolution of networks over time and for identifying sudden changes and shocks.

Methods for comparing networks range from comparison of summary statistics to sophisticated alignment-based approaches [Kuchaiev and Pržulj, 2011, Neyshabur et al., 2013, Mamano and Hayes, 2017]<sup>AEW: [References reordered.]</sup>. Alignment based approaches aim to identify structurally similar regions of networks by finding a mapping between network nodes that maximizes the overlap between the networks. This mapping information is often the principal aim of such methods which for instance in the context of protein-protein interactions, could be used to infer the biological function and structure of proteins in different organisms [Singh et al., 2008, Kuchaiev and Pržulj, 2011, Neyshabur et al., 2013, Mamano and Hayes, 2017]<sup>AEW: [References reordered.]</sup>. Although network alignment is *NP*-hard in general several efficient heuristics have emerged over recent years [Hashemifar and Xu, 2014, Mamano and Hayes, 2017]<sup>AEW: [References reordered.]</sup>. On the other hand, alignment-free methods in the form of network similarity measures and network distances aim to quantify the overall similarity of networks on the basis of network features. Alignment free methods typically are computationally less expensive than alignment based methods and can effectively compare large sets of networks. Network comparison measures have many applications such as goodness of fit tests of random graph models of real world networks [Pržulj, 2007, Rito et al., 2010]<sup>AEW: [Reference added.]</sup> and the tracking the evolution of network time series [Kossinets and Watts, 2006]. Network comparison measures have also attracted increasing attention in the field of machine learning, where they are mostly referred to as graph kernels, with applications for example in personalized medicine e.g. [Borgwardt et al., 2007], computer vision and drug discovery e.g. [Wale et al., 2008].

Real-world networks can be very large and are often inhomogeneous, which makes the problem of network comparison challenging, especially when networks differ significantly in terms of size and density. In this paper, we address this problem by introducing a new network comparison methodology that is aimed at comparing networks according to their common organizational principles.

The observation that the degree distribution of many real world networks is highly right skewed and in many cases approximately follows a power law has been very influential in the development of network science [Barabási and Albert, 1999]. Consequently, it has become widely accepted that the shape of the degree distribution (for example, binomial vs power law) is indicative of the generating mechanism underlying the network. In this paper, we formalize this idea by introducing a measure that captures the shape of distributions. The measure emerges from the requirement that a metric between forms of distributions should be invariant under rescalings and translations of the observables. Based on this measure, we then introduce a new network comparison methodology, which we call *NetEmd*.

Although our methodology is applicable to almost any type of feature that can be associated to nodes or edges of a graph, we focus mainly on distributions of small connected subgraphs, also known as graphlets. Graphlets form the basis of many of the state of the art network comparison methods [Pržulj, 2007, Ali et al., 2014, Yaveroglu et al., 2014] and hence using graphlet based features allows for a comparative assessment of the presented methodology. Moreover, certain subgraph patterns, called network motifs [Milo et al., 2002, Masoudi-Nejad et al., 2012], occur much more frequently in many real world networks than is expected on the basis of pure chance. Network motifs are considered to be basic building blocks of networks that contribute to the function of the network by performing modular tasks and have therefore been conjectured to be favoured by natural selection. This is supported by the observation that network motifs are largely conserved within classes of networks [Milo et al., 2004, Wegner, 2014].

Our methodology provides an effective tool for comparing networks even when networks differ

significantly in size and density, which is the case in most applications. The methodology performs well on a wide variety of networks ranging from chemical compounds having as few as 10 nodes to internet network with tens of thousands of nodes. The method achieves state of the art performance even when based on a rather restricted set of features, specifically distributions of graphlets up to size 3. Graphlets of size 3 can be enumerated very efficiently and hence when based on these features the method scales favourably to networks with millions and even billions of nodes. The method also behaves well under the network sub-sampling [Holmes et al., 2004, Yaveroglu et al., 2014, Bhattacharyya et al., 2015, Ali et al., 2016]<sup>AEW</sup>: [References reordered.]. The methodology further meets the needs of researchers from a variety of fields, from the social sciences to the biological and life sciences, by being computationally efficient and simple to implement.

We test the presented methodology in a large number of settings, starting with clustering synthetic and real world networks, where we find that the presented methodology outperforms state of the art graphlet-based network comparison methods in clustering networks of different sizes and densities. We then test the more fine grained properties of *NetEmd* using data sets that represent evolving networks at different points in time. Finally, we test whether *NetEmd* can predict functional categories of networks by exploring machine learning applications and find that classifiers based on *NetEmd* outperform state-of-the-art graph classifiers on several benchmark data sets.

## 2. A measure for comparing shapes of distributions

Here we build on the idea that the information encapsulated in the shape of the degree distribution and other network properties reflects the topological organization of the network. From an abstract point of view we think of the shape of a distribution as a property that is invariant under linear deformations i.e. translations and re-scalings of the axis. For example, a Gaussian distribution always has its characteristic bell curve shape regardless of its mean and standard deviation. Consequently, we postulate that any metric that aims to capture the similarity of shapes should be invariant under such linear transformations of its inputs.

Based on these ideas we define the following measure between distributions  $p$  and  $q$  that are supported on  $\mathbb{R}$  and have non-zero, finite variances:

$$EMD^*(p, q) = \inf_{c \in \mathbb{R}} (EMD(\tilde{p}(\cdot + c), \tilde{q}(\cdot))), \quad (2.1)$$

where  $EMD$  is the earth mover's distance and  $\tilde{p}$  and  $\tilde{q}$  are the distributions obtained by rescaling  $p$  and  $q$  to have variance 1. More precisely,  $\tilde{p}$  is the distribution obtained from  $p$  by the transformation  $x \rightarrow \frac{x}{\sigma(p)}$ , where  $\sigma(p)$  is the standard deviation of  $p$ . Intuitively,  $EMD$  (also known as the 1st Wasserstein metric [Runber et al., 1998]) can be thought of as the minimal work, i.e. mass times distance, needed to "transport" the mass of one distribution onto the other. For probability distributions  $p$  and  $q$  with support in  $\mathbb{R}$  and bounded absolute first moment, the  $EMD$  between  $p$  and  $q$  is given by  $EMD(p, q) = \int_{-\infty}^{\infty} |F(x) - G(x)| dx$ , where  $F$  and  $G$  are the cumulative distribution functions of  $p$  and  $q$  respectively.

In principle,  $EMD$  in Equation (2.1) can be replaced by almost any other probability metric  $d$  to obtain a corresponding metric  $d^*$ . Here we choose  $EMD$  because it is well suited to comparing shapes, as shown by its many applications in the area of pattern recognition and image retrieval [Runber et al., 1998]. Moreover, we found that  $EMD$  produces superior results to classical  $L^1$  and Kolmogorov distances, especially for highly irregular distributions that one frequently encounters in real world networks.

For two networks  $G$  and  $G'$  and given network feature  $t$ , we define the corresponding *NetEmd* <sub>$t$</sub>  measure by:

$$NetEmd_t(G, G') = EMD^*(p_t(G), p_t(G')), \quad (2.2)$$

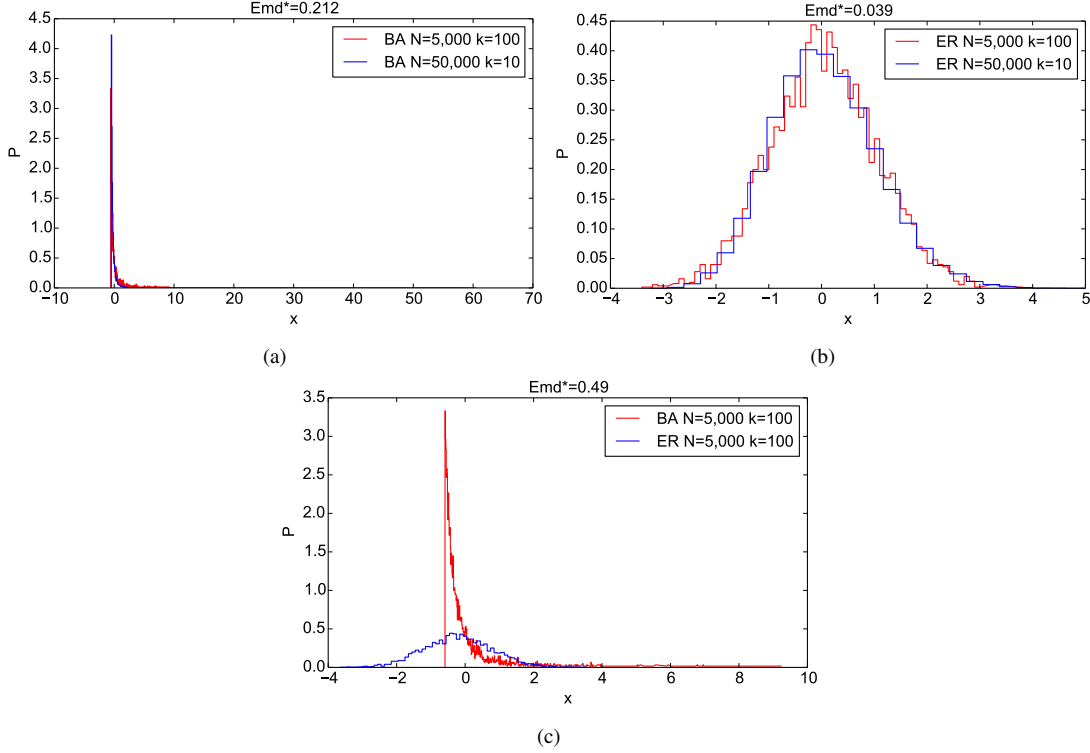


FIG. 1: Plots of rescaled and translated degree distributions for Barabasi-Albert (BA) and Erdős-Rényi (ER) models with  $N$  nodes and average degree  $k$ : a) BA  $N = 5,000$ ,  $k = 100$  vs BA  $N = 50,000$ ,  $k = 10$ . b) ER  $N = 5,000$ ,  $k = 100$  vs ER  $N = 50,000$ ,  $k = 10$ . c) BA  $N = 5,000$ ,  $k = 100$  vs ER  $N = 5,000$ ,  $k = 100$ . The  $EMD^*$  distances between the degree distribution of two BA or ER models with quite different values of  $N$  and  $k$  are smaller than the  $EMD^*$  distance between the degree distribution of a BA and ER model when the number of nodes and average degree are equal.

where  $p_t(G)$  and  $p_t(G')$  are the distributions of  $t$  on  $G$  and  $G'$  respectively.  $NetEmd_t$  can be shown to be a pseudometric between graphs for any feature  $t$  (see Appendix Sec. B), that is it is non-negative, symmetric and satisfies the triangle inequality. Figure 1 gives examples where  $t$  is taken to be the degree distribution, and  $p_t(G)$  is the degree distribution of  $G$ .

Metrics based on a single feature might not be effective in capturing the structural differences between networks as two networks that agree in terms of a single feature such as the degree distribution might still differ significantly in terms of other structures. For instance, there are many generation mechanisms that produce networks with power-law type degree distributions but differ significantly in other aspects. Consequently, measures that are based on the comparison of multiple features can be expected to be more effective at identifying structural differences between networks than measures that are based on a single feature  $t$ , because for two networks to be considered similar they must show similarity across multiple features. Hence, for a given set  $T = \{t_1, t_2, \dots, t_m\}$  of network features, we define the  $NetEmd$

measure corresponding to  $T$  simply as:

$$NetEmd_T(G, G') = \frac{1}{m} \sum_{j=1}^m NetEmd_{t_j}(G, G'). \quad (2.3)$$

Although *NetEmd* can in principle be based on any set  $T$  of network features to which one can associate distributions, we initially consider only features that are based on distributions of small connected subgraphs, also known as graphlets. Graphlets form the basis of many state of the art network comparison methods and hence allow for a comparative assessment of the proposed methodology that is independent of the choice of features.

First, we consider graphlet degree distributions (*GDDs*) [Pržulj, 2007] as our set of features. For a given graphlet  $m$ , the graphlet degree of a node is the number of graphlet- $m$  induced subgraphs that are attached to the node. One can distinguish between the different positions the node can have in  $m$ , which correspond to the automorphism orbits of  $m$ , see Figure 2. We initially take the set of 73 *GDDs* corresponding to graphlets up to size 5 to be the default set of inputs, for which we denote the metric as *NetEmd<sub>G5</sub>*.

Later we also explore alternative definitions of subgraph distributions based on ego networks, as well as the effect of varying the size of subgraphs considered in the input. Finally, we consider the eigenvalue spectra of the graph Laplacian and the normalized graph Laplacian as inputs.

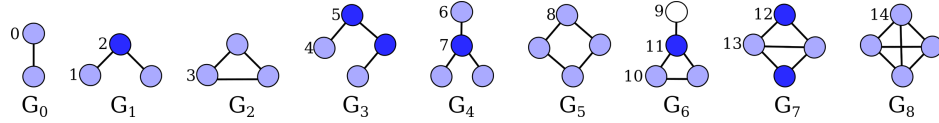


FIG. 2: Graphlets on two to four nodes. The different shades in each graphlet represent different automorphism orbits, numbered from 0 to 14.

### 3. Results

In order to give a comparative assessment of *NetEmd*, we consider other graphlet based network comparison methods, namely *GDDA* [Pržulj, 2007], *GCD* [Yaveroglu et al., 2014] and *Netdis* [Ali et al., 2014]. These represent the most effective alignment-free network comparison methodologies in the existing literature. While *GDDA* directly compares distributions of graphlets up to size 5 in a pairwise fashion, *GCD* is based on comparing rank correlations between graphlet degrees. Here we consider both default settings of *GCD* [Yaveroglu et al., 2014], namely *GCD11*, which is based on a non-redundant subset of 11 graphlets up to size 4, and *GCD73* which uses all graphlets up to size 5. *Netdis* differs from *GDDA* and *GCD* in that it is based on subgraph counts in ego-networks of nodes. Another important distinction is that *Netdis* first centers these raw counts by comparing them to the counts that could be expected under a particular null model before computing the final statistics. In our analysis, we consider two null models: an Erdős-Rényi random graph and a duplication divergence graph [Vázquez et al., 2003] which has a scale-free degree distribution as well as a high clustering coefficient. We denote these two variants as *Netdis<sub>ER</sub>* and *Netdis<sub>SF</sub>*, respectively.

We initially report results of a single variant *NetEmd<sub>G5</sub>* that is based on graphlets up to size 5 in order to provide a comparison between the different methods that is independent of the choice of features. Results obtained using different feature sets are discussed in Sec. 3.3.

### 3.1 Clustering synthetic and real world networks

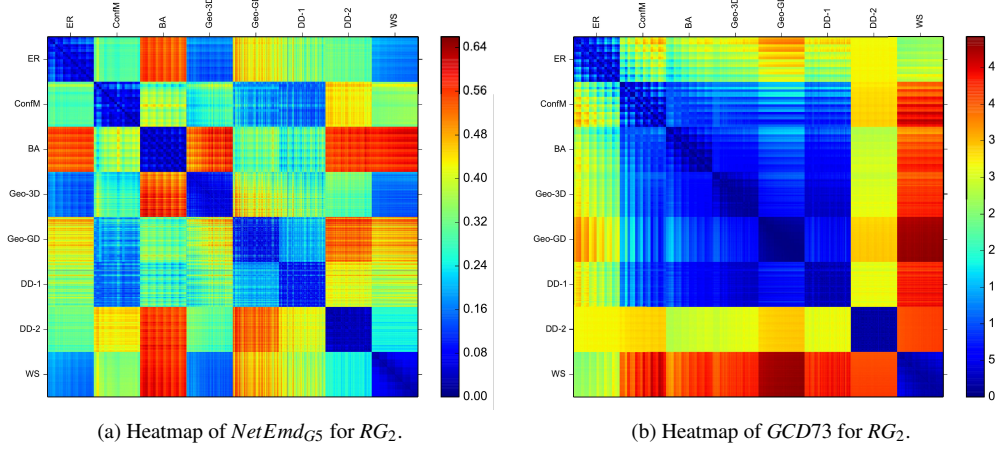
We start with the classical setting of network comparison where the task is to identify groups of structurally similar networks. The main challenge in this setting is to identify structurally similar networks even though they might differ substantially in terms of size and density.

Given a set  $S = \{G_1, G_2, \dots, G_n\}$  of networks consisting of disjoint classes  $C = \{c_1, c_2, \dots, c_m\}$  one would like a network comparison measure  $d$  to position networks from the same class closer to each other when compared to networks from other classes. Given a network  $G$ , this can be measured in terms of the empirical probability  $P(G)$  that  $d(G, G_1) < d(G, G_2)$  where  $G_1$  is a randomly selected network from the same class as  $G$  (excluding itself) and  $G_2$  is a randomly selected network from outside the class of  $G$  and  $d$  is the network comparison statistic. Consequently, the performance over the whole data set is measured in terms of the quantity  $\bar{P} = \frac{1}{|S|} \sum_{G \in S} P(G)$ . It can be shown that  $P(G)$  is equivalent to the area under the receiver operator characteristic curve of a binary classifier that for a given network  $G$  classifies the  $k$  nearest neighbours of  $G$  with respect to  $d$  as being in the same class as  $G$  (See Appendix H). Hence, a measure that positions networks randomly has an expected  $\bar{P}$  of 0.5 whereas  $\bar{P} = 1$  corresponds to perfect separation between classes. Other measures are discussed in the Appendix. Conclusions reached in this paper hold regardless of which of these performance measures one uses.

We first test *NetEmd* on synthetic networks corresponding to realizations of eight random graph models, namely the Erdős-Rényi random graphs [Erdős and Rényi, 1960], the Barabási Albert preferential attachment model [Barabási and Albert, 1999], two duplication divergence models [Vázquez et al., 2003, Ispolatov et al., 2005], the geometric gene duplication model [Higham et al., 2008], 3D geometric random graphs [Penrose, 2003], the configuration model [Molloy and Reed, 1995], and Watts-Strogatz small world networks [Watts and Strogatz, 1998] (see Sec. G.1 in the Appendix for details).

For synthetic networks we consider three experimental settings of increasing difficulty, starting with the task of clustering networks that have same size  $N$  and average degree  $k$  according to generating mechanism - a task that is relevant in a model selection setting. For this we generate 16 data sets, which collectively we call  $RG_1$ , corresponding to combinations of  $N \in \{1250, 2500, 5000, 10000\}$  and  $k \in \{10, 20, 40, 80\}$ , each containing 10 realizations per model, i.e. 80 networks. This is an easier problem than clustering networks of different sizes and densities, and in this setting we find that the  $\bar{P}$  scores (see Table 3c) of top performing measures tend to be within one standard deviation of each other. We find that *NetEmd*<sub>G5</sub> and *GCD*<sub>73</sub> achieve the highest scores, followed by *GCD*<sub>11</sub> and *Netdis*<sub>SF</sub>.

Having established that *NetEmd* is able to differentiate networks according to generating mechanism, we move on to the task of clustering networks of different sizes and densities. For this we generate two data sets:  $RG_2$  in which the size  $N$  and average degree  $k$  are increased independently in linear steps to twice their initial value ( $N \in \{2000, 3000, 4000\}$  and  $k \in \{20, 24, 28, 32, 36, 40\}$ ) and  $RG_3$  in which the size and average degree are increased independently in multiples of 2 to 8 times their initial value ( $N \in \{1250, 2500, 5000, 10000\}$  and  $k \in \{10, 20, 40, 80\}$ ). In  $RG_3$ , the number of nodes and average degrees of the networks both vary by one order of magnitude, and therefore clustering according to model type is challenging. Both  $RG_2$  and  $RG_3$  contain 10 realizations per model parameter so that these contain  $3 \times 6 \times 8 \times 10 = 1440$  and  $4 \times 4 \times 8 \times 10 = 1280$  networks, respectively. Finally, we consider a data set consisting of networks from 10 different classes of real world networks (RWN) as well as a data set from [Ali et al., 2014] that consists of real world and synthetic networks from the larger collection compiled by Onnela *et al.* [Onnela et al., 2012]. Since the generation mechanisms of real world networks are generally unknown the 'ground truth' of real world data sets is based on the assumption that networks from a certain domain are structurally more similar to each other than to networks from different domains.



Dataset	$NetEmd_{G5}$	$Netdis_{ER}$	$Netdis_{SF}$	$GCD11$	$GCD73$	$GDDA$
Synthetic Networks						
$RG_1$	<b><math>0.997 \pm 0.003</math></b>	$0.981 \pm 0.013$	$0.986 \pm 0.011$	$0.992 \pm 0.012$	$0.996 \pm 0.005$	$0.952 \pm 0.056$
$RG_2$	<b>0.988</b>	0.897	0.919	0.976	0.976	0.956
$RG_3$	<b>0.925</b>	0.790	0.800	0.872	0.861	0.812
RWN	<b>0.942</b>	0.898	0.866	0.898	0.906	0.745
Onnela et al.	<b>0.890</b>	0.832	0.809	0.789	0.819	0.783

(c)  $\bar{P}$  values for different network measures on data sets of synthetic and real world networks.

FIG. 3: a) and b) show the heatmaps of pairwise distances on  $RG_2$  ( $N \in \{2000, 3000, 4000\}$  and  $k \in \{20, 24, 28, 32, 36, 40\}$ ) according to  $NetEmd_{G5}$  and  $GCD73$ , respectively. In the heat map, networks are ordered from top to bottom in the following order: model, average degree and node count. The heatmap of  $NetEmd$  shows eight clearly identifiable blocks on the diagonal corresponding to different generative models while the heatmap of  $GCD73$  shows signs of off-diagonal mixing. c)  $\bar{P}$  values for various comparison measures for data sets of synthetic and real world networks. For  $RG_1$  we calculated the value of  $\bar{P}$  for each of the 16 sub-data sets. The table shows the average and standard deviation of the  $\bar{P}$  values obtained over these 16 sub-data sets.

We find that  $NetEmd_{G5}$  outperforms all of the other three methods at clustering networks of different sizes and densities on all data sets. The difference can also be seen in the heatmaps of  $NetEmd_{G5}$  and  $GCD73$ , the second best performing method for  $RG_2$ , given in Figures 3a and 3b. While the heatmap of  $NetEmd_{G5}$  shows eight clearly identifiable blocks on the diagonal corresponding to different generative models, the heatmap of  $GCD73$  shows signs of off-diagonal mixing. The difference in performance becomes even more pronounced on more challenging data sets, i.e. on  $RG_3$  (see Fig. A.6 in the Appendix) and the Onnela *et al.* data set.

### 3.2 Time ordered networks

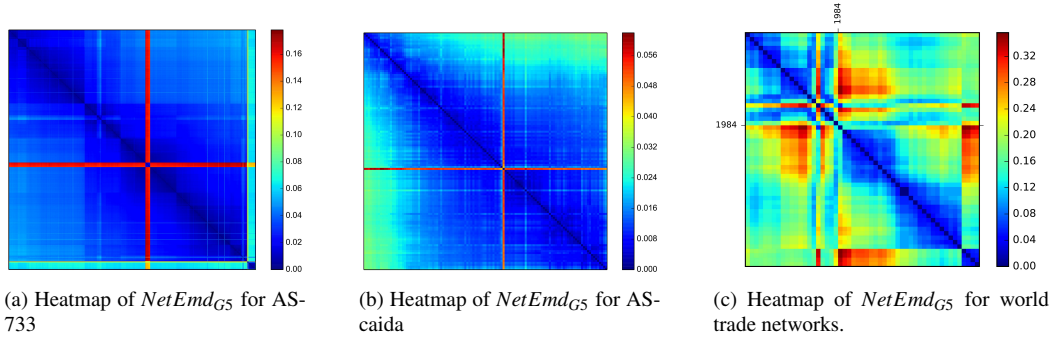
A network comparison measure should ideally not only be able to identify groups of similar networks but should also be able to capture structural similarity at a finer local scale. To study the behaviour of  $NetEmd$  at a more local level, we consider data sets that represent a system measured at different



points in time. Since it is reasonable to assume that such networks evolve gradually over time towards their final state they offer a sensible experimental setting for testing the local properties of network comparison methodologies.

We consider two data sets named AS-caida and AS-733 [Leskovec et al., 2005] that represent the topology of the Internet at the level of autonomous systems and a third data set that consists of bilateral trade flows between countries for the years 1962–2014 [Feenstra et al., 2005, United-Nations-Statistics-Division, 2015]. Both edges and nodes are added and deleted over time in all three data sets. As was noted in [Leskovec et al., 2005] the time ranking in evolving networks is reflected to a certain degree in simple summary statistics. Hence, recovering the time ranking of evolving networks should be regarded as a test of consistency rather than an evaluation of performance.

In order to minimize the dependence of our results on the algorithm that is used to rank networks, we consider four different ways of ranking networks based on their pairwise distances as follows. We assume that either the first or last network in the time series is given. Rankings are then constructed in a step-wise fashion. At each step one either adds the network that is closest to the last added network (Algorithm 1), or adds the network that has smallest average distance to all the networks in the ranking constructed so far (Algorithm 2). The performance of a measure in ranking networks is then measured in terms of Kendall’s rank correlation coefficient  $\tau$  between the true time ranking and the best ranking obtained by any of the 4 methods.



Dataset	$NetEmd_{G5}$	$Netdis_{ER}$	$Netdis_{SF}$	$GCD11$	$GCD73$	$GDDA$
AS-733	0.874	0.867	<b>0.933</b>	0.763	0.770	0.740
AS-caida	0.890	0.844	0.849	<b>0.897</b>	0.878	0.870
World Trade	<b>0.821</b>	0.666	0.388	0.380	0.567	0.649

(d) Kendall’s  $\tau$  between the true time ranking and rankings inferred from network comparison methodologies.

FIG. 4: (a), (b) & (c) Heatmaps of  $NetEmd_{G5}$  for networks representing the internet at the level of autonomous systems networks and world trade networks. The date of measurement increases from left to right/ top to bottom.  $NetEmd_{G5}$  accurately captures the evolution over time in all three data sets by positioning networks that are close in time closer to each other resulting in a clear signal along the diagonal. (d) Kendall’s rank correlation coefficient between the true time ranking and rankings inferred from different network comparison measures.

We find that  $NetEmd_{G5}$  successfully recovers the time ordering for all three data sets, as can be seen



in the time ordered heatmaps given in Figure 4 which all show clear groupings along the diagonal. The red regions in the two internet data sets correspond to outliers which can also be identified as sudden jumps in summary statistics e.g. the number of nodes. The two large clusters in the heatmap of world trade networks (Figure 4c) coincide with a change in the data gathering methodology in 1984 [Feenstra et al., 2005]. Although  $NetEmd_{G5}$  comes second to  $Netdis_{SF}$  on AS-733 and to  $GCD11$  on AS-caida,  $NetEmd_{G5}$  has the highest overall score and is the only measure that achieves consistently high scores on all three data sets.

### 3.3 $NetEmd$ based on different sets of inputs

We examine the effect of reducing the size of graphlets considered in the input of  $NetEmd$ , which is also relevant from a computational point of view, since enumerating graphlets up to size 5 can be challenging for very large networks. We consider variants based on the graphlet degree distributions of graphlets up to size 3 and 4, which we denote as  $NetEmd_{G3}$  and  $NetEmd_{G4}$ . We also consider  $NetEmd_{DD}$  which is based only on the degree distribution as a baseline. Results are given in Table 1.

We find that reducing the size of graphlets from 5 to 4 does not significantly decrease the performance of  $NetEmd$  and actually produces better results on three data sets ( $RG_3$ , Real world and Onnela et al.). Even when based on only graphlets up to size 3, i.e. just edges, 2-paths and triangles,  $NetEmd$  outperforms all other non- $NetEmd$  methods that we tested on at least 6 out of 8 data sets.

Given that the complexity of enumerating graphlets up to size  $s$  in a network on  $N$  nodes having maximum degree  $k_{max}$  is  $O(Nk_{max}^{s-1})$ ,  $NetEmd_{G4}$  offers an optimal combination of performance and computational efficiency in most cases. The even less computationally costly  $NetEmd_{G3}$  scales favourably even to networks of billions of edges for which enumerating graphlets of size 4 can be computationally prohibitive. This opens the door for comparing very large networks which are outside the reach of current methods while still retaining state of the art performance. Furthermore,  $NetEmd$  behaves well under the bootstrap which samples nodes and uses the only the graphlet degrees of these nodes as inputs; see for example [Holmes et al., 2004, Yaveroglu et al., 2014, Bhattacharyya et al., 2015, Ali et al., 2016] <sup>AEW:</sup> [References reordered.]. Sub-sampling can be leveraged to further improve computational efficiency of  $NetEmd$  (see Appendix D).

We find that in some cases restricting the set of inputs actually leads to an increase in the performance of  $NetEmd$ . This indicates that not all graphlet distributions are equally informative in all settings [Maugis et al., 2017]. Consequently, identifying (learning) which graphlet distributions contain the most pertinent information for a given task might lead to significant improvements in performance. Such generalizations can be incorporated into  $NetEmd$  in a straightforward manner, for instance by modifying the sum in Equation (2.3) to incorporate weights.  $NetEmd$  is ideally suited for such metric learning [Xing et al., 2003] type generalizations since it constructs an individual distance for each graphlet distribution. Moreover, such single feature  $NetEmd$  measures are in many cases highly informative even on their own. For instance  $NetEmd_{DD}$ , which only uses the degree distribution, outperforms the non- $NetEmd$  measures we tested individually on more than half the data sets we considered.

We also considered counts of graphlets up to size 4 in 1-step ego networks of nodes ( $NetEmd_{E4}$ ) [Ali et al., 2014] as an alternative way of capturing subgraph distributions, for which we denote the measure as  $NetEmd_{E4}$ . Although we find that  $NetEmd_{E4}$  achieves consistently high scores variants based on graphlet degree distributions tend to perform better on most data sets.

Finally, we consider spectral distributions of graphs as a possible alternative to graphlet based features. The spectra of various graph operators are closely related to topological properties of graphs [Mohar et al., 1991, Chung, 1997, Banerjee and Jost, 2008] <sup>AEW:</sup> [References reordered.] and have been

widely used to characterize and compare graphs [Wilson and Zhu, 2008, Gu et al., 2016]<sup>AEW: [References reordered.]</sup>. We used the spectra of the graph Laplacian and normalized graph Laplacian as inputs for *NetEmd* for which we denote the measure as *NetEmd<sub>S</sub>*. For a given graph the Laplacian is defined as  $L = D - A$  where  $A$  is the adjacency matrix of the graph and  $D$  is the diagonal matrix whose diagonal entries are the node degrees. The normalized Laplacian  $\hat{L}$  is defined as  $D^{-\frac{1}{2}}LD^{-\frac{1}{2}}$ . Given the eigenvalue distributions  $S(L)$  and  $S(\hat{L})$  of  $L$  and  $\hat{L}$  we define *NetEmd<sub>S</sub>* to be  $\frac{1}{2}(NetEmd_{S(L)} + NetEmd_{S(\hat{L})})$ .

We find that in general *NetEmd<sub>S</sub>* performs better in clustering random graphs of different sizes and densities when compared to graphlet based network comparison measures. However, on the RWN and Onnela et al. data sets graphlet based *NetEmd* measures tend to perform better than the spectral variant which can be attributed to the prevalence of network motifs in real world networks, giving graphlet based measures an advantage. The spectral variant is also outperformed on the time ordering of data sets which in turn might be a result of the sensitivity of graph spectra to small changes in the underlying graph [Wilson and Zhu, 2008].

Data set	<i>NetEmd<sub>G3</sub></i>	<i>NetEmd<sub>G4</sub></i>	<i>NetEmd<sub>G5</sub></i>	<i>NetEmd<sub>E4</sub></i>	<i>NetEmd<sub>S</sub></i>	<i>NetEmd<sub>DD</sub></i>	Best Other
<i>RG<sub>1</sub></i>	0.989±0.008	0.995±0.005	<b>0.997±0.003</b>	0.993±0.004	0.992±0.007	0.957±0.024	0.996±0.005 ( <i>GCD73</i> )
<i>RG<sub>2</sub></i>	0.982	0.987	0.988	0.983	<b>0.992</b>	0.944	0.976( <i>GCD73</i> )
<i>RG<sub>3</sub></i>	0.940	0.941	0.925	0.947	<b>0.972</b>	0.902	0.872( <i>GCD11</i> )
RWN	<b>0.952</b>	0.950	0.942	0.933	0.933	0.907	0.906( <i>GCD73</i> )
Onnela et al.	0.892	<b>0.898</b>	0.890	0.892	0.858	0.867	0.832( <i>Netdis<sub>ER</sub></i> )
AS-733	0.808	0.874	0.874	0.922	0.855	0.928	<b>0.933</b> ( <i>Netdis<sub>SF</sub></i> )
AS-caida	<b>0.898</b>	0.892	0.890	0.820	0.780	0.821	0.897( <i>GCD11</i> )
World Trade	0.697	0.785	<b>0.821</b>	0.665	0.430	0.358	0.666( <i>Netdis<sub>ER</sub></i> )

Table 1: Results for different variants of *NetEmd* based on distributions of graphlets up to size 3 and 4 (*NetEmd<sub>G3</sub>* and *NetEmd<sub>G4</sub>*), counts of graphlets up to size 4 in 1-step ego networks of nodes (*NetEmd<sub>E4</sub>*), eigenvalue spectra of Laplacian operators (*NetEmd<sub>S</sub>*) and the degree distribution (*NetEmd<sub>DD</sub>*). Values in bold indicate that a measure achieves the highest score among all measures considered in the manuscript. For *RG<sub>1</sub>* we calculate the value of  $\bar{P}$  for each of the 16 sub-data sets. The table shows the average and standard deviation of the  $\bar{P}$  values obtained over these 16 sub-data sets.

### 3.4 Functional classification of networks

One of the primary motivations in studying the structure of networks is to identify topological features that can be related to the function of a network. In the context of network comparison this translates into the problem of finding metrics that can identify functionally similar networks based on their topological structure.

In order to test whether *NetEmd* can be used to identify functionally similar networks, we use several benchmarks from the machine learning literature where graph similarity measures, called graph kernels, have been intensively studied over the past decade. In the context of machine learning the goal is to construct classifiers that can accurately predict the class membership of unknown graphs.

We test *NetEmd* on benchmark data sets representing social networks [Yanardag and Vishwanathan, 2015] consisting of Reddit posts, scientific collaborations and ego networks in the Internet Movie Database (IMDB). The Reddit data sets Reddit-Binary, Reddit-Multi-5k and Reddit-Multi-12k consist of networks representing Reddit threads where nodes correspond to users and two users are connected whenever one responded to the other’s comments. While for the Reddit-Binary data sets the task is to classify networks into discussion based and question/answer based communities, in the data sets Reddit-Multi-5k and Reddit-Multi-12k the task is to classify networks according to their subreddit categories.

COLLAB is a data set consisting of ego-networks of scientists from the fields High Energy Physics, Condensed Matter Physics and Astro Physics and the task is to determine which of these fields a given researcher belongs to. Similarly, the data sets IMDB-Binary and IMDB-Multi represent collaborations between film actors derived from the IMDB and the task is to classify ego-networks into different genres i.e. action and romance in the case of IMDB-Binary and comedy, action and Sci-Fi genres in the case of IMDB-Multi.

We use C - support vector machine (C-SVM) [Cortes and Vapnik, 1995] classifiers with a Gaussian kernel  $K(G, G') = \exp(-\frac{NetEmd(G, G')^2}{2\alpha^2})$ , where  $\alpha$  is a free parameter to be learned during training. Performance evaluation is carried out by 10 fold cross validation, where at each step of the validation 9 folds are used for training and 1 fold for evaluation. Free parameters of classifiers are learned via 10 fold cross validation on the training data only. Finally, every experiment is repeated 10 fold and average prediction accuracy and standard deviation are reported.

Kernel	Reddit-Binary	Reddit-Multi-5k	Reddit-Multi-12k	COLLAB	IMDB-Binary	IMDB-Multi
<i>NetEmd<sub>G5</sub></i>	<b>92.67 ± 0.30</b>	<b>54.61 ± 0.18</b>	<b>48.09 ± 0.21</b>	<b>79.32 ± 0.27</b>	66.99 ± 1.19	41.45 ± 0.70
<i>NetEmd<sub>S</sub></i>	88.59 ± 0.35	53.05 ± 0.34	44.45 ± 0.18	<b>79.05 ± 0.20</b>	<b>71.68 ± 0.88</b>	<b>46.06 ± 0.50</b>
DGK	78.04 ± 0.39	41.27 ± 0.18	32.22 ± 0.10	73.09 ± 0.25	66.96 ± 0.56	44.55 ± 0.52
GK	77.34 ± 0.18	41.01 ± 0.17	31.82 ± 0.08	72.84 ± 0.28	65.87 ± 0.98	43.89 ± 0.38
RF	88.7 ± 1.99	50.9 ± 2.07	42.7 ± 1.28	76.5 ± 1.68	<b>72.4 ± 4.68</b>	<b>47.8 ± 3.55</b>
PCSN	86.30 ± 1.58	49.10 ± 0.70	41.32 ± 0.42	72.60 ± 2.15	<b>71.00 ± 2.29</b>	<b>45.23 ± 2.84</b>

Table 2: 10 fold cross validation accuracies of Gaussian kernels based on *NetEmd* measures using the distributions of graphlets up to size 5 (*NetEmd<sub>G5</sub>*) and Laplacian spectra (*NetEmd<sub>S</sub>*) and other graph kernels, namely the deep graphlet kernels (DGK)[Yanardag and Vishwanathan, 2015] and the graphlet kernel (GK) [Shervashidze et al., 2009]. We also consider alternatives to support vector machines classifiers, namely the random forest classifiers (RF) introduced in [Barnett et al., 2016] and convolutional neural networks (PCSN) [Niepert et al., 2016]. Values in bold correspond to significantly higher scores, which are scores with t-test p-values less than 0.05 when compared to the highest score.

Table 2 gives classification accuracies obtained using *NetEmd* measures based on graphlets up to size five (*NetEmd<sub>G5</sub>*) and spectra of Laplacian operators (*NetEmd<sub>S</sub>*) on the data sets representing social networks. We compare *NetEmd* based kernels to graphlet kernels [Shervashidze et al., 2009] and deep graphlet kernels [Yanardag and Vishwanathan, 2015] as well as two non-SVM classifiers namely the random forest (RF) classifier introduced in [Barnett et al., 2016] and the convolutional neural network based classifier introduced in [Niepert et al., 2016]. Short descriptions of the classifiers is given in the Appendix (Sec F.2).

On the Reddit data sets and the COLLAB data set, *NetEmd<sub>G5</sub>* significantly outperforms other state-of-the-art graph classifiers. On the other hand, we find that *NetEmd<sub>G5</sub>* performs poorly on the IMDB data sets. This can be traced back to the large number of complete graphs present in the IMDB data sets: 139 out of the 1000 graphs in IMDB-Binary and 789 out of 1500 graphs in IMDB-Multi are complete graphs which mainly correspond to ego-networks of actors having acted only in a single film. By definition, *NetEmd<sub>G5</sub>* cannot distinguish between complete graphs of different sizes since all graphlet degree distributions are concentrated on a single value in complete graphs. The spectral variant *NetEmd<sub>S</sub>* is not affected by this and we find that *NetEmd<sub>S</sub>* is either on par with or outperforms the other non-*NetEmd* graph classifiers on all six data sets.

We also tested *NetEmd* on benchmark data sets representing chemical compounds and protein structures. Unlike the social network data sets, in these data sets nodes and edges are labeled to reflect domain

specific knowledge such as atomic number, amino acid type and bond type. Although *NetEmd*, in contrast to the other graph kernels, does not rely on domain specific knowledge in the form of node or edge labels, we found that *NetEmd* outperforms many of the considered graph kernels coming only second to the Weisfeiler-Lehman [Shervashidze et al., 2011] type kernels in terms of overall performance (see Appendix E).

#### 4. Discussion

Starting from basic principles, we have introduced a general network comparison methodology, *NetEmd*, that is aimed at capturing common generating processes in networks. We tested *NetEmd* in a large variety of experimental settings and found that *NetEmd* successfully identifies similar networks at multiple scales even when networks differ significantly in terms of size and density, generally outperforming other graphlet based network comparison measures. Even when based only on graphlets up to size 3 (i.e. edges, 2-paths and triangles), *NetEmd* has performance comparable to the state of the art, making *NetEmd* feasible even for networks containing billions of edges and nodes.

By exploring machine learning applications we showed that *NetEmd* captures topological similarity in a way that relates to the function of networks and outperforms state-of-the art graph classifiers on several graph classification benchmarks.

Although we only considered variants of *NetEmd* that are based on distributions of graphlets and spectra of Laplacian operators in this paper, *NetEmd* can also be applied to other graph features in a straightforward manner. For instance, distributions of paths and centrality measures might capture larger scale properties of networks and their inclusion into *NetEmd* might lead to a more refined measure.

#### Data availability

The source code for *NetEmd* is freely available at: <http://opig.stats.ox.ac.uk/resources>

#### Acknowledgements

This work was in part supported by EPSRC grant EP/K032402/1 (A.W, G.R, C.D and R.G) and EPSRC grants EP/G037280/1 and EP/L016044/1 (C.D). L.O acknowledges the support of Colciencias through grant 568. R.G. acknowledges support from the COST Action CA15109 and is currently supported by a Dame Kathleen Ollerenshaw Research Fellowship. C.D. and G.R. acknowledge the support of the Alan Turing Institute (grant EP/NS10129/1).

We thank Xiaochuan Xu and Martin O'Reilly for useful discussions.

#### A. Implementation

##### A.1 Graphlet distributions.

In the main paper, both the graphlet degree distribution and graphlet counts in 1-step ego networks were used as inputs for *NetEmd*.

**GRAPHLET DEGREE DISTRIBUTIONS.** The graphlet degree [Pržulj, 2007] of a node specifies the number of graphlets (small induced subgraphs) of a certain type the node appears in, while distinguishing between different positions the node can have in a graphlet. Different positions within a graphlet correspond to the orbits of the automorphism group of the graphlet. Among graphs on two to four nodes,

there are 9 possible graphs and 15 possible orbits. Among graphs on two to five nodes there are 30 possible graphs and 73 possible orbits.

**GRAPHLET DISTRIBUTIONS BASED ON EGO-NETWORKS.** Another way of obtaining graphlet distributions is to consider graphlet counts in ego-networks [Ali et al., 2014]. The  $k$ -step ego-network of a node  $i$  is defined as the subgraph induced on all the nodes that can be reached from  $i$  (including  $i$ ) in less than  $k$  steps. For a given  $k$ , the distribution of a graphlet  $m$  in a network  $G$  is then simply obtained by counting the occurrence of  $m$  as an induced subgraph in the  $k$ -step ego-networks of each individual node.

### A.2 Step-wise implementation

In this paper, for integer valued network features such as graphlet based distributions, we base our implementation on the probability distribution that corresponds to the histogram of feature  $t$  with bin width 1 as  $p_t(G)$ .  $NetEmd$  can also be defined on the basis of discrete empirical distributions i.e. distributions consisting of point masses (See Section C).

Here we summarise the calculation of the  $NetEmd_T(G, G')$  distance between networks  $G$  and  $G'$  (with  $N$  and  $N'$  nodes respectively), based on the comparison of the set of local network features  $T = \{t_1, \dots, t_m\}$  of graphlet degrees corresponding to graphlets up to size  $k$ .

1. First one computes the graphlet degree sequences corresponding to graphlets up to size  $k$  for networks  $G$  and  $G'$ . This can be done efficiently using the algorithm ORCA [Hočevár and Demšar, 2014]. For the graphlet degree  $t_1$  compute a histogram across all  $N$  nodes of  $G$  having bins of width 1 of which the centers are at their respective values. This histogram is then normalized to have total mass 1. We then interpret the histogram as the (piecewise continuous) probability density function of a random variable. This probability density function is denoted by  $p_{t_1}(G)$ . The standard deviation of  $p_{t_1}(G)$  is then computed, and is used to rescale the distribution so that it has variance 1. This distribution is denoted by  $\widetilde{p_{t_1}(G)}$ .
2. Repeat the above step for network  $G'$ , and denote the resulting distribution by  $\widetilde{p_{t_1}(G')}$ . Now compute

$$NetEmd_{t_1}^*(G, G') = \inf_{c \in \mathbb{R}} (EMD(\widetilde{p_{t_1}(G)}(\cdot + c), \widetilde{p_{t_1}(G')}(\cdot))).$$

In practice, this minimisation over  $c$  is computed using a suitable optimization algorithm. In our implementation we use the Brent-Dekker algorithm [Brent, 1971] with an error tolerance of 0.00001 and with the number of iterations upper bounded by 150.

3. Repeat the above two steps for the network features  $t_2, \dots, t_m$  and compute

$$NetEmd_T(G, G') = \frac{1}{m} \sum_{j=1}^m NetEmd_{t_j}^*(G, G').$$

### A.3 Example: $EMD^*$ for Gaussian distributions

Suppose that  $p$  and  $q$  are  $N(\mu_1, \sigma_1^2)$  and  $N(\mu_2, \sigma_2^2)$  distributions, respectively. Then

$$\begin{aligned}
EMD^*(p, q) &= \inf_{c \in \mathbb{R}} \left( EMD(\tilde{p}(\cdot + c), \tilde{q}(\cdot)) \right) \\
&= EMD\left(\tilde{p}\left(\cdot - \frac{\mu_1}{\sigma_1} + \frac{\mu_2}{\sigma_2}\right), \tilde{q}(\cdot)\right) \\
&= EMD(\tilde{q}(\cdot), \tilde{q}(\cdot)) = 0.
\end{aligned}$$

Here we used that if  $X \sim N(\mu_1, \sigma_1^2)$  and  $Y \sim N(\mu_2, \sigma_2^2)$ , then  $\frac{X}{\sigma_1} + c \sim N(\frac{\mu_1}{\sigma_1} + c, 1)$  and  $\frac{Y}{\sigma_2} \sim N(\frac{\mu_2}{\sigma_2}, 1)$ , and these two distributions are equal if  $c = \frac{\mu_1}{\sigma_1} - \frac{\mu_2}{\sigma_2}$ .

#### A.4 Spectral NetEmd

When using spectra of graph operators, which take real values instead of the integer values one has in the case of graphlet distributions, we use the empirical distribution consisting of point masses for computing *NetEmd*. For more details see Section C of this appendix.

#### A.5 Computational complexity

The computational complexity of graphlet based comparison methods is dominated by the complexity of enumerating graphlets. For a network of size  $N$  and maximum degree  $d$ , enumerating all connected graphlets up to size  $m$  has complexity  $O(Nd^{m-1})$ , while counting all graphlets up to size  $m$  in all  $k$ -step ego-networks has complexity  $O(Nd^{k+m-1})$ . Because most real world networks are sparse, graphlet enumeration algorithms tends to scale more favourably in practice than the worst case upper bounds given above.

In the case of spectral measures, the most commonly used algorithms for computing the eigenvalue spectrum have complexity  $O(N^3)$ . Recent results show that the spectra of graph operators can be approximated efficiently in  $O(N^2)$  time [Thüne, 2013].

Given the distribution of a feature  $t$ , computing  $EMD_t^*(G, G')$  has complexity  $O(k(s+s')\log(s+s'))$ , where  $s$  and  $s'$  are the number of different values  $t$  takes in  $G$  and  $G'$  respectively and  $k$  is the maximum number function calls of the optimization algorithm used to align the distributions. For node based features such as graphlet distributions, the worst case complexity is  $O(k(N(G) + N(G'))\log(N(G) + N(G')))$ , where  $N(G)$  is the number of nodes of  $G$ , since the number of different values  $t$  can take is bounded by the number of nodes.

We use the combinatorial graphlet enumeration algorithm ORCA [Hočevár and Demšar, 2014] for enumerating graphlets. Typical runtimes for enumerating all graphlets up to size 5 are can be found in Table A.3 along with runtimes for *NetEmd<sub>G5</sub>* given the corresponding graphlet degree distributions. Enumerating graphlets of size 5 for networks of size around 1.000 nodes tends to take several minutes allowing data sets consisting of several hundreds of such networks to be analyzed on a single CPU in a matter of hours. Larger data sets containing networks of the order of 10.000 nodes or larger are computationally challenging since enumerating graphlets of size 5 for such networks can take several hours [Hočevár and Demšar, 2014]. Consequently, computing *NetEmd<sub>G5</sub>* on data sets such as *RG<sub>3</sub>* and *Reddit - Multi - 12k* are computationally challenging tasks that can take up to 3 days on a 24 CPU cluster.

Number of nodes Average degrees	N=250 (10,20,50,100)	N=1,000 (10,20,50,100)	N=5,000 (10,20,50,100)	N=10,000 (10,20,50,100)
Erdős-Rényi model ORCA	(0.02s,0.17s,3.10s,16.5s)	(0.09s,0.65s,9.68s,96.6s)	(0.40s,2.38s,40.7s,490s)	(1.36s,7.21s,82s,858s)
Barabási Albert ORCA	(0.06s,0.37s,3.29s,34.7s)	(0.38s,2.29s,25.9s,184s)	(3.53s,15.4s,272s,2366s)	(6.66s,57.5s,786s,5870s)
<i>NetEmd<sub>G5</sub></i> $t_{min}-t_{max}$	0.27s-0.72s	0.66s-2.53s	1.63s-10.5s	2.58s-18.0s

Table A.3: Runtimes for enumerating graphlets up to size 5 using ORCA for Erdős-Rényi and a Barabási Albert random graphs and computation times for evaluating *NetEmd<sub>G5</sub>* given the graphlet degree distributions. For *NetEmd<sub>G5</sub>* the range  $t_{min}-t_{max}$  obtained over all pairwise comparisons of all networks of a given size is shown. Experiments were performed on a single core of an Intel i7-6700HQ processor. ORCA is implemented in C++ and *NetEmd<sub>G5</sub>* is implemented in Python and Cython.

## B. Proof that *NetEmd* is a distance measure

We begin by stating a definition. A *pseudometric* on a set  $X$  is a non-negative real-valued function  $d : X \times X \rightarrow [0, \infty)$  such that, for all  $x, y, z \in X$ ,

1.  $d(x, x) = 0$ ;
2.  $d(x, y) = d(y, x)$  (symmetry);
3.  $d(x, z) \leq d(x, y) + d(y, z)$  (triangle inequality).

If Condition 1 is replaced by the condition that  $d(x, y) = 0 \iff x = y$  then  $d$  defines a *metric*. Note that this requirement can only be satisfied by a network comparison measure that is based on a complete set of graph invariants and hence network comparison measures in general will not satisfy this requirement.

**PROPOSITION** Let  $M$  denote the space of all real-valued probability measures supported on  $\mathbb{R}$  with finite, non-zero variance. Then the  $EMD^*$  distance between probability measures,  $\mu_X$  and  $\mu_Y$  in  $M$  defined by

$$EMD^*(\mu_X, \mu_Y) = \inf_{c \in \mathbb{R}} EMD(\tilde{\mu}_X(\cdot), \tilde{\mu}_Y(\cdot + c)),$$

defines a pseudometric on the space of probability measures  $M$ .

**PROOF** We first note that if  $\mu_X \in M$  then  $\tilde{\mu}_X(\cdot + c) \in M$  for any  $c \in \mathbb{R}$ . Let us now verify that  $EMD^*$  satisfies all properties of a pseudometric. Clearly, for any  $\mu_X \in M$ , we have  $0 \leq EMD^*(\mu_X, \mu_X) \leq EMD(\tilde{\mu}_X(\cdot), \tilde{\mu}_X(\cdot)) = 0$ , and so  $EMD^*(\mu_X, \mu_X) = 0$ . Symmetry holds, since for, any  $\mu_X$  and  $\mu_Y$  in  $M$ ,

$$\begin{aligned}
EMD^*(\mu_X, \mu_Y) &= \inf_{c \in \mathbb{R}} EMD(\tilde{\mu}_X(\cdot), \tilde{\mu}_Y(\cdot + c)) \\
&= \inf_{c \in \mathbb{R}} EMD(\tilde{\mu}_Y(\cdot + c), \tilde{\mu}_X(\cdot)) \\
&= \inf_{c \in \mathbb{R}} EMD(\tilde{\mu}_Y(\cdot), \tilde{\mu}_X(\cdot + c)) \\
&= EMD^*(\mu_Y, \mu_X).
\end{aligned}$$



Finally, we verify that  $EMD^*$  satisfies the triangle inequality. Suppose  $\mu_X, \mu_Y$  and  $\mu_Z$  are probability measures from the space  $M$ , then so are  $\tilde{\mu}_X(\cdot + a), \tilde{\mu}_Y(\cdot + b)$  for any  $a, b \in \mathbb{R}$ . Since  $EMD$  satisfies the triangle inequality, we have, for any  $a, b \in \mathbb{R}$ ,

$$EMD(\tilde{\mu}_X(\cdot + a), \tilde{\mu}_Y(\cdot + b)) \leq EMD(\tilde{\mu}_X(\cdot + a), \tilde{\mu}_Z(\cdot)) + EMD(\tilde{\mu}_Y(\cdot + b), \tilde{\mu}_Z(\cdot)).$$

Since the above inequality holds for all  $a, b \in \mathbb{R}$ , we have that

$$\begin{aligned} EMD^*(\mu_X, \mu_Y) &= \inf_{c \in \mathbb{R}} EMD(\tilde{\mu}_X(\cdot + c), \tilde{\mu}_Y(\cdot)) \\ &= \inf_{a, b \in \mathbb{R}} EMD(\tilde{\mu}_X(\cdot + a), \tilde{\mu}_Y(\cdot + b)) \\ &\leq \inf_{a, b \in \mathbb{R}} [EMD(\tilde{\mu}_X(\cdot + a), \tilde{\mu}_Z(\cdot)) + EMD(\tilde{\mu}_Y(\cdot + b), \mu_Z(\cdot))] \\ &= \inf_{a \in \mathbb{R}} [EMD(\tilde{\mu}_X(\cdot + a), \tilde{\mu}_Z(\cdot)) + \inf_{b \in \mathbb{R}} EMD(\tilde{\mu}_Y(\cdot + b), \tilde{\mu}_Z(\cdot))] \\ &= \inf_{a \in \mathbb{R}} EMD(\tilde{\mu}_X(\cdot + a), \tilde{\mu}_Z(\cdot)) + \inf_{b \in \mathbb{R}} EMD(\tilde{\mu}_Y(\cdot + b), \tilde{\mu}_Z(\cdot)) \\ &= EMD^*(\mu_X, \mu_Z) + EMD^*(\mu_Y, \mu_Z), \end{aligned}$$

as required. We have thus verified that  $EMD^*$  satisfies all properties of a pseudometric.  $\square$

### C. Generalization of $EMD^*$ to point masses

Although in the case of graphlet based features we based our implementation of *NetEmd* on probability distribution functions that correspond to normalized histograms having bin width 1 *NetEmd* can also be based on empirical distributions consisting of collections of point masses located at the observed values.

The definition of  $EMD^*$  can be generalized to include distributions of zero variance, i.e. unit point masses. Mathematically, the distribution of a point mass at  $x_0$  is given by the Dirac measure  $\delta_x(x_0)$ . Such distributions are frequently encountered in practice since some graphlets do not occur in certain networks.

First, we note that unit point masses are always mapped onto unit point masses under rescaling operations. Moreover, for a unit point mass  $\delta_x(x_0)$  we have that  $\inf_{c \in \mathbb{R}} (EMD(\tilde{p}(\cdot + c), \delta_x(x_0))) = \inf_{c \in \mathbb{R}} (EMD(\tilde{p}(\cdot + c), \delta_x(kx_0)))$  for all  $p \in M$  and  $k > 0$ . Consequently,  $EMD^*$  can be generalized to include unit point masses in a consistent fashion by always rescaling them by 1:

$$EMD^*(p, q) = \inf_{c \in \mathbb{R}} (EMD(\hat{p}(\cdot + c), \hat{q})),$$

where  $\hat{p} = \tilde{p}$  (as in Eq. 2.1) if  $p$  has a non-zero variance, and  $\hat{p} = p$  if  $p$  has variance zero.

### D. Sub-sampling

*NetEmd* is well suited for network sub-sampling [Yaveroglu et al., 2014, Bhattacharyya et al., 2015, Ali et al., 2016]<sup>AEW</sup>. [References reordered.]. In the case of graphlets the sub-sampling procedure consist of sampling nodes and using the graphlet degrees corresponding to these nodes only as inputs for *NetEmd*.

Figure A.5 shows the  $\bar{P}$  scores for variants of *NetEmd* on a set of synthetic networks and the Onnela et al. data set. We find that the performance of *NetEmd* is stable under sub-sampling and that in general using a sample of only 10% of the nodes produces results comparable to the case where all nodes are used.

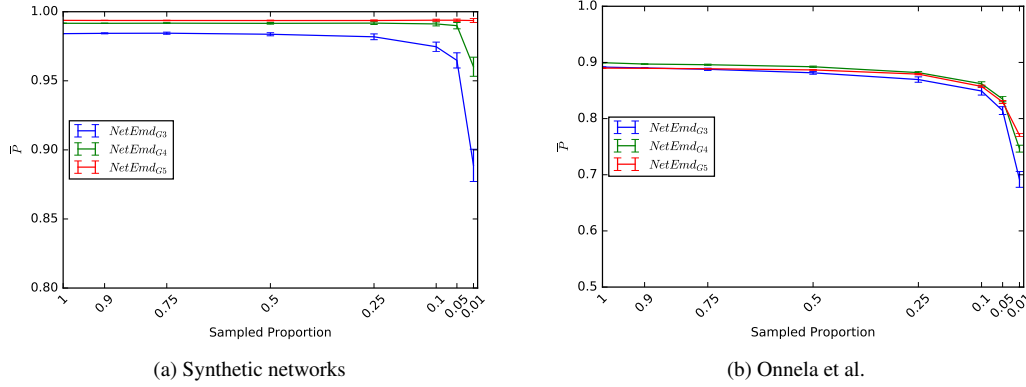


FIG. A.5: The  $\bar{P}$  values for different variants of *NetEmd* under sub-sampling for a) a set of 80 synthetic networks coming from eight different random graph models with 2500 nodes and average degree 20, b) for the Onnela et al. data set showing the average and standard deviation over 50 experiments for each sampled proportion. Note that the performance of *NetEmd* under sub-sampling is remarkably stable and is close to optimal even when only 10% of nodes are sampled. For synthetic networks we find that the stability of *NetEmd* increases as the size of the graphlets used in the input is increased.

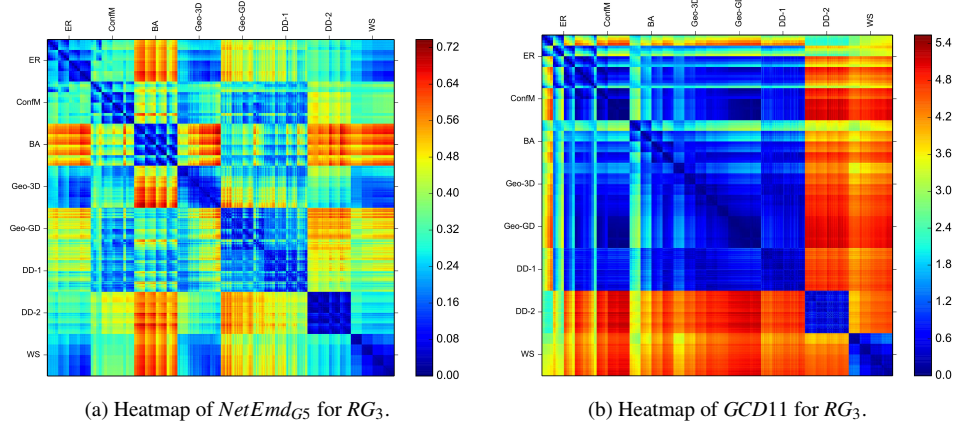


FIG. A.6: a) and b) show the heatmaps of pairwise distances on  $RG_3$  ( $N \in \{1250, 2500, 5000, 10000\}$  and  $k \in \{10, 20, 40, 80\}$ ) according to *NetEmd*<sub>G5</sub> and next best performing measure *GCD11*, respectively. In the heat map, networks are ordered from top to bottom in the following order: model, average degree and node count. Although we observe some degree of off diagonal mixing the heatmap of *NetEmd* still shows 8 diagonal blocks corresponding to different generative models in contrast to the heat map of *GCD11*.

### E. Results for data sets of chemical compounds and proteins

We also tested *NetEmd* on benchmark data sets representing chemical compounds (MUTAG, NCI1 and NCI109) and protein structures (ENZYMES and D&D). MUTAG [Debnath et al., 1991] is a data

Kernel	MUTAG	NCI1	NCI109	ENZYMES	D & D
<i>NetEmd<sub>G5</sub></i>	83.71 ±1.16	78.59±0.28	76.71±0.34	46.55±1.25	78.01 ±0.38
<i>NetEmd<sub>S</sub></i>	83.30 ±1.20	77.36±0.38	76.14±0.27	42.75±0.78	76.74 ±0.43
WL subtree	82.05±0.36	82.19 ±0.18	82.46 ±0.24	52.22±1.26	79.78 ±0.36
WL edge	81.06±1.95	84.37±0.30	84.49±0.20	53.17±2.04	77.95±0.70
WL shortest path	83.78±1.46	84.55±0.36	83.53±0.30	59.05±1.05	79.43±0.55
Ramon & Gärtner	85.72±0.49	61.86±0.27	61.67±0.21	13.35±0.87	57.27±0.07
p-random walk	79.19±1.09	58.66±0.28	58.36±0.94	27.67±0.95	66.64±0.83
Random walk	80.72±0.38	64.34±0.27	63.51±0.18	21.68±0.94	71.70±0.47
Graphlet count	75.61±0.49	66.00±0.07	66.59±0.08	32.70±1.20	78.59±0.12
Shortest path	87.28±0.55	73.47±0.11	73.07±0.11	41.68±1.79	78.45±0.26

Table A.4: 10 fold cross validation accuracies of Gaussian kernels based on *NetEmd<sub>G5</sub>* and *NetEmd<sub>S</sub>* and other kernels reported in [Shervashidze et al., 2011].

set of 188 chemical compounds that are labelled according to their mutagenic effect on *Salmonella typhimurium*. NCI1 and NCI109 represent sets of chemical compounds which are labelled for their activity against non-small cell lung cancer and ovarian cancer cell lines, respectively [Wale et al., 2008]. Nodes and edges in MUTAG, NCI1 and NCI109 are labeled by atomic number and bond type, respectively. ENZYMES and D&D [Borgwardt et al., 2005] consist of networks representing protein structures at the level of tertiary structure and amino acids respectively. While networks in ENZYMES are classified into six different enzyme classes, networks in D&D are classified according to whether or not they correspond to an enzyme. Nodes in ENZYMES are labelled according to structural element type and according to amino acid types in D&D.

Classification accuracies obtained using *NetEmd* on the data sets of chemical compounds and protein structures are given in Table A.4, along with results for other graph kernels reported in [Shervashidze et al., 2011]. For a detailed description of these kernels we refer to [Shervashidze et al., 2011] and the references therein. Note that, in contrast to all other kernels in Table A.4, *NetEmd* does not use any domain specific knowledge in the form of node or edge labels. Node and edge labels are highly informative for all five classification tasks - as shown in [Sugiyama and Borgwardt, 2015].

On MUTAG, *NetEmd* achieves an accuracy that is comparable to the Weisfeiler-Lehman (WL) shortest path kernel, but is outperformed by the shortest path kernel and the kernel by Ramon & Gärtner. While on NCI1, NCI109 and ENZYMES, *NetEmd* is outperformed only by WL kernels, on D&D *NetEmd* achieves a classification accuracy that is comparable to the best performing kernels. Notably, on D&D *NetEmd* also outperforms the vector model by Dobson and Doig [Dobson and Doig, 2003] (classification accuracy:  $76.86 \pm 1.23$ ) which is based on 52 physical and chemical features without using domain specific knowledge i.e. solely based on graph topology.

## F. Graph classifiers

### F.1 Implementation of C-SVMs

Following the procedure in [Shervashidze et al., 2011] we use 10-fold cross validation with a C-SVM [Cortes and Vapnik, 1995] to test classification performance. We use the python package scikit-learn [Pedregosa et al., 2011] which is based on libsvm [Chang and Lin, 2011]. The  $C$  - value of the C-SVM and the  $\alpha$  for the Gaussian kernel is tuned independently for each fold using training data from that fold only. We consider values  $\{2^{-7}, 2^{-6}, \dots, 2^7\}$  for  $C$  and the  $\{2^{-7}, 2^{-6}, \dots, 2^7\}$  multiples of the median of

the distance matrix for  $\alpha$ . Each experiment is repeated 10 times, and average prediction accuracies and their standard deviations are reported.

We also note that the Gaussian NetEmd kernel is not necessarily positive semidefinite for all values of  $\alpha$  [Jayasumana et al., 2015]. The implication is that the C-SVM might converge to a stationary point that is not always guaranteed to be a global optimum. Although there exist alternative algorithms [Luss and d’Aspremont, 2008] for training C-SVMs with indefinite kernels which might result in better classification accuracy, here we chose to use the standard libsvm-algorithm in order to ensure a fair comparison between kernels. For a discussion of support vector machines with indefinite kernels see [Haasdonk, 2005].

## F.2 Other kernels and classifiers

**F.2.1 Graph kernels.** Graph kernels are in general based on counting various types substructures in graphs such as walks (p-random walk, Random walk), paths (Shortest path) and sub-trees (Ramon & Gärtner). Given the count vectors of the substructures of interest the corresponding kernel is defined as their inner product.

**F.2.2 The graphlet kernel and the deep graphlet kernel.** The graphlet kernel and the deep graphlet kernel use the normalized counts of graphlets on  $k$  nodes, including disconnected graphlets, as features. The deep graphlet kernel involves the additional step of learning similarities between graphlets based on the edit distance. In the case of the social network data sets  $k=7$ .

**F.2.3 Weisfeiler Lehman kernels.** The Weisfeiler Lehman (WL) kernels are based on the Weisfeiler Lehman label propagation procedure in which a node is iteratively labeled by the labels of its neighbours. Given a base kernel the corresponding WL kernel is obtained by combining the base kernel over several iterations of the WL procedure. The WL-subtree, WL-edge and WL-shortest path kernels have base kernels that use the following counts as features: node labels, labeled edges and shortest paths with labeled end points, respectively.

**F.2.4 The random forest classifier.** The random forest classifier in [Barnett et al., 2016] is based on the following features: number of nodes, number of edges, average degree, degree assortativity, number of triangles, and the global clustering coefficient.

**F.2.5 PCSN.** The convolutional neural network classifier of Niepert et al. is based on learning feature representations of graphs that correspond to locally connected regions of networks. The approach first identifies a sequence of nodes for which neighbourhood graphs are created and then maps these onto a vector space representation.

## G. Detailed description of data sets and models

### G.1 Synthetic networks and random graph models

$RG_1$  consists of 16 sub data sets corresponding to combinations of  $N \in \{1250, 2500, 5000, 10000\}$  and  $k \in \{10, 20, 40, 80\}$  containing 10 realizations for each model so that each of the 16 sub data set contains 80 networks.

In  $RG_2$  the size  $N$  and average degree  $k$  are increased independently in linear steps to twice their initial value ( $N \in \{2000, 3000, 4000\}$  and  $k \in \{20, 24, 28, 32, 36, 40\}$ ) and contains 10 realizations per model parameter combination, resulting in a data set of  $3 \times 6 \times 8 \times 10 = 1440$  networks.

In  $RG_3$  the size  $N$  and average degree  $k$  are increased independently in multiples of 2 to 8 times their initial value ( $N \in \{1250, 2500, 5000, 10000\}$  and  $k \in \{10, 20, 40, 80\}$ ) and again contains 10 realizations per model parameter combination, resulting in a data set of  $4 \times 4 \times 8 \times 10 = 1280$  networks. The models are as follows.

**G.1.1 *The Erdős-Rényi model.*** We consider the Erdős-Rényi (ER) model [Erdős and Rényi, 1960]  $G(N, m)$  where  $N$  is the number of nodes and  $m$  is the number of edges. The edges are chosen uniformly at random without replacement from the  $\binom{N}{2}$  possible edges.

**G.1.2 *The configuration model.*** Given a graphical degree sequence, the configuration model creates a random graph that is drawn uniformly at random from the space of all graphs with the given degree sequence. The degree sequence of the configuration models used in the paper is taken to be degree sequence of a duplication divergence model that has the desired average degree.

**G.1.3 *The Barabási Albert preferential attachment model.*** In the Barabási-Albert model [Barabási and Albert, 1999] a network is generated starting from a small initial network to which nodes of degree  $m$  are added iteratively and the probability of connecting the new node to an existing node is proportional to the degree of the existing node.

**G.1.4 *Geometric random graphs.*** Geometric random graphs [Gilbert, 1961] are constructed under the assumption that the nodes in the network are embedded into a  $D$  dimensional space, and the presence of an edge depends only on the distance between the nodes and a given threshold  $r$ . The model is constructed by placing  $N$  nodes uniformly at random in an  $D$ -dimensional square  $[0, 1]^D$ . Then edges are placed between any pair of nodes for which the distance between them is less or equal to the threshold  $r$ . We use  $D = 3$  and set  $r$  to be the threshold that results in a network with the desired average degree, while the distance is the Euclidean distance.

**G.1.5 *The geometric gene duplication model.*** The geometric gene duplication model is a geometric model [Higham et al., 2008] in which the nodes are distributed in 3 dimensional Euclidean space  $\mathbb{R}^3$  according to the following rule. Starting from an small initial set of nodes in three dimensions, at each step a randomly chosen node is selected and a new node is placed at random within a Euclidean distance  $d$  of this node. The process is repeated until the desired number of nodes is reached. Nodes within a certain distance  $r$  are then connected. We fix  $r$  to obtain the desired average degree.

**G.1.6 *The duplication divergence model of Vázquez et al..*** The duplication divergence model of Vázquez et al. [Vázquez et al., 2003] is defined by the following growing rules: (1) Duplication: A node  $v_i$  is randomly selected and duplicated ( $v'_i$ ) along with all of its interactions. An edge between  $v_i$  and  $v'_i$  is placed with probability  $p$ . (2) Divergence: For each pair of duplicated edges  $\{(v_i, v_k); (v'_i, v_k)\}$ ; one of the duplicated edges is selected uniformly at random and then deleted with probability  $q$ . This process is followed until the desired number of nodes is reached. In our case we fix  $p$  to be 0.05 and adjust  $q$  through a grid search to obtain a network that on average has the desired average degree.

Data set	#Networks	$N_{min}$	Median( $N$ )	$N_{max}$	$E_{min}$	Median( $E$ )	$E_{max}$	$d_{min}$	Median( $d$ )	$d_{max}$
RWN	167	24	351	62586	76	2595	824617	7.55e-05	0.0163	0.625
Onnela et al.	151	30	918	11586	62	2436	232794	4.26e-5	0.0147	0.499
AS-caida	122	8020	22883	26475	18203	46290	53601	1.48e-4	1.78e-4	5.66e-4
AS-733	732	493	4180.5	6474	1234	8380.5	13895	6.63e-4	9.71e-4	1.01e-2
World Trade Networks	53	156	195	242	5132	7675	18083	0.333	0.515	0.625
Reddit-Binary	2000	6	304.5	3782	4	379	4071	5.69e-4	8.25e-3	0.286
Reddit-Multi-5k	4999	22	374	3648	21	422	4783	6.55e-4	6.03e-3	0.091
Reddit-Multi-12k	11929	2	280	3782	1	323	5171	5.69e-4	8.27e-3	1.0
COLLAB	5000	32	52	492	60	654.5	40120	0.029	0.424	1.0
IMDB-Binary	1000	12	17	136	26	65	1249	0.095	0.462	1.0
IMDB-Multi	1500	7	10	89	12	36	1467	0.127	1.0	1.0
MUTAG	188	10	17.5	28	10	19	33	0.082	0.132	0.222
NCI1	4110	3	27	111	2	29	119	0.0192	0.0855	0.667
NCI109	4127	4	26	111	3	29	119	0.0192	0.0862	0.5
ENZYMES	600	2	32	125	1	60	149	0.0182	0.130	1.0
D&D	1178	30	241	5748	63	610.5	14267	8.64e-4	0.0207	0.2

Table A.5: Summary statistics of data sets  $N$ ,  $E$  and  $d$  stand for the number of nodes, number of edges and edge density, respectively.

**G.1.7 The duplication divergence of Ispolatov et al.** The duplication divergence model of Ispolatov et al. [Ispolatov et al., 2005] starts with an initial network consisting of a single edge and then at each step a random node is chosen for duplication and the duplicate is connected to each of the neighbours of its parent with probability  $p$ . We adjust  $p$  to obtain networks that have on average the desired average degree.

**G.1.8 The Watts-Strogatz model.** The Watts-Strogatz model, [Watts and Strogatz, 1998] creates graphs that interpolate between regular graphs and ER graphs. The model starts with a ring of  $n$  nodes in which each node is connected to its  $k$ -nearest neighbours in both directions of the ring. Each edges is rewired with probability  $p$  to a node which is selected uniformly at random. While  $k$  is adjusted to obtain networks having the desired average degree we take  $p$  to be 0.05.

## G.2 Real world data sets

Summary statistics of the data sets are given in Table A.5.

**G.2.1 Real world networks from different classes (RWN).** We compiled a data set consisting of 10 different classes of real world networks: social networks, metabolic networks, protein interaction networks, protein structure networks, food webs, autonomous systems networks of the internet, world trade networks, airline networks, peer to peer file sharing networks and scientific collaboration networks. Although in some instances larger versions of these data sets are available, we restrict the maximum number of networks in a certain class to 20 by taking random samples of larger data sets in order to avoid scores being dominated by larger network classes.

The class of social networks consists of 10 social networks from the Pajek data set which can be found at <http://vlado.fmf.uni-lj.si/pub/networks/data/default.htm> (June 12th 2015) (Networks: 'bkfrat', 'bkham', 'dolphins', 'kaptailS1', 'kaptailS2', 'kaptailT1', 'kaptailT2', 'karate', 'lesmis', 'prison') and a sample of 10 Facebook networks from [Traud et al., 2012] (Networks: 'Bucknell39', 'Duke14', 'Harvard1', 'MU78', 'Maine59', 'Rice31', 'UC61', 'UCLA26', 'UVA16', 'Yale4'). The class of metabolic networks consists of 20 networks taken from [Jeong et al., 2000] (Networks: 'AB', 'AG', 'AP', 'AT', 'BS', 'CE', 'CT', 'EF', 'HI', 'MG', 'MJ', 'ML', 'NG', 'OS', 'PA', 'PN', 'RP', 'TH', 'TM', 'YP'). The class of protein interaction networks consists of 6 networks from BIOGRID

[Stark et al., 2006] (Arabidopsis thaliana, Caenorhabditis elegans, Drosophila melanogaster, Homo sapiens, Mus musculus and Saccharomyces cerevisiae downloaded: October 2015) and 5 networks from HINT [Das and Yu, 2012] (Arabidopsis thaliana, Caenorhabditis elegans, Drosophila melanogaster, Homo sapiens and Mus musculus (Version: June 1 2014)) and the protein interaction network of Escheria coli by Rajagopala et al. [Rajagopala et al., 2014]. The class of protein structure networks consists of a sample of 20 networks from the data set D&D (Networks: 20, 119, 231, 279, 335, 354, 355, 369, 386, 462, 523, 529, 597, 748, 833, 866, 990, 1043, 1113, 1157). The class of food webs consists of 20 food webs from the Pajek data set: <http://vlado.fmf.uni-lj.si/pub/networks/data/default.htm> (June 10th 2015) (Networks: 'ChesLower', 'ChesMiddle', 'ChesUpper', 'Chesapeake', 'CrystalC', 'CrystalD', 'Everglades', 'Florida', 'Michigan', 'Mondego', 'Narragan', 'StMarks', 'baydry', 'baywet', 'cypdry', 'cypwet', 'gramdry', 'gramwet', 'mangdry', 'mangwet'). The class of internet networks consists of 10 randomly chosen networks from AS-733 [Leskovec et al., 2005] (Networks: '1997/11/12', '1997/12/28', '1998/01/01', '1998/06/06', '1998/08/13', '1998/12/04', '1999/03/30', '1999/04/17', '1999/06/18', '1999/08/30') and 10 randomly chosen networks from AS-caida [Leskovec et al., 2005] (Networks: '2004/10/04', '2006/01/23', '2006/03/27', '2006/07/10', '2006/09/25', '2006/11/27', '2007/01/15', '2007/04/30', '2007/05/28', '2007/09/24'). Both datasets are from SNAP [Jure and Krevl, 2014](June 1 2016). The class of world trade networks is a sample of 20 networks of the larger data set considered in [Feenstra et al., 2005, United-Nations-Statistics-Division, 2015] (Networks: 1968, 1971, 1974, 1975, 1976, 1978, 1980, 1984, 1989, 1992, 1993, 1996, 1998, 2001, 2003, 2005, 2007, 2010, 2011, 2012). The airline networks were derived from the data available at: <http://openflights.org/> (June 12 2015). For this we considered the 50 largest airlines from the database in terms of the number of destinations that the airline serves. For each airline a network is obtained by the considering all airports that are serviced by the airlines which are connected whenever there is direct flight between a pair of nodes. We then took a sample of 20 networks from this larger data set (Airline codes of the networks: 'AD', 'AF', 'AM', 'BA', 'DY', 'FL', 'FR', 'JJ', 'JL', 'MH', 'MU', 'NH', 'QF', 'SU', 'SV', 'U2', 'UA', 'US', 'VY', 'ZH'). The class of peer to peer networks consist of 9 networks of the Gnutella file sharing platform measured at different dates which are available at [Jure and Krevl, 2014]. The scientific collaboration networks consists of 5 networks representing different scientific disciplines which were obtained from [Jure and Krevl, 2014] (June 1 2015).

**G.2.2 Onnela et al. data set.** The Onnela et al. data set consists of all undirected and unweighted networks from the larger collection analysed in [Onnela et al., 2012]. A complete list of networks and class membership can be found in the supplementary information of [Ali et al., 2014].

**G.2.3 Time ordered data sets.** The data sets AS-caida and AS-733 each represent the internet measured at the level of autonomous systems at various points in time. Both data sets were downloaded from [Jure and Krevl, 2014](June 1 2015).

The World Trade Networks data set is based on the data set [Feenstra et al., 2005] for the years 1962-2000 and on UN COMTRADE [United-Nations-Statistics-Division, 2015] for the years 2001-2015. Two countries are connected in the network whenever they import or export a commodity from a each other within the given calendar year. The complete data set was downloaded from : <http://atlas.media.mit.edu/en/resources/data/> on July 12 2015.

**G.2.4 Machine learning benchmarks.** A short description of the social networks datasets was given in the main text. A more detailed description can be found in [Yanardag and Vishwanathan, 2015].



The social network data sets were downloaded from <https://ls11-www.cs.tu-dortmund.de/staff/morris/graphkerneldatasets> on September 2 2016.

A short short description of the chemical compound and protein structure data sets was given in Section E. A more detailed description of the data set can be found in [Shervashidze et al., 2011]. These data sets were downloaded from: <https://www.bsse.ethz.ch/mlcb/research/machine-learning/graph-kernels.html> on June 12 2016.

## H. $\bar{P}$ and area under ROC

We assume that our data set  $S = \{G_1, G_2, \dots, G_n\}$  of networks consists of disjoint classes  $C = \{c_1, c_2, \dots, c_m\}$ . For a given distance measure  $d$  and given a network  $G \in S$  we want to calculate the empirical probability  $P(G)$  that  $d(G, G_1) < d(G, G_2)$  where  $G_1$  is a randomly selected network from the same class as  $G$  (excluding itself) and  $G_2$  is a randomly selected network from outside the class of  $G$ . Given measure  $d$ ,  $P(G)$  is simply given by:

$$P(G) = \frac{1}{(|S| - |C(G)|)(|C(G)| - 1)} \sum_{G_1 \in C_G \setminus \{G\}} \sum_{G_2 \in S \setminus C_G} I(d(G, G_1) < d(G, G_2)), \quad (\text{A.1})$$

where  $C_G$  denotes the set of networks in the same class of  $G$  and  $I$  is the indicator function. Now we compare this to the area under ROC of the classifier  $kNN(G)$  that classifies the  $k$ -nearest neighbours of  $G$  to be in the same class  $G$ . Let the  $k^{th}$  nearest neighbour of  $G$  according to  $d$  be  $G_k$ . For a given  $k$  the true positive ( $TP$ ) and false positive ( $FP$ ) rates can be written as:

$$TP(k) = \frac{1}{(|C(G)| - 1)} \sum_{G_1 \in C_G \setminus \{G\}} I(d(G, G_1) \leq d(G, G_k)). \quad (\text{A.2})$$

$$FP(k) = \frac{1}{(|S| - |C(G)|)} \sum_{G_2 \in S \setminus C_G} I(d(G, G_2) \leq d(G, G_k)). \quad (\text{A.3})$$

The ROC curve (i.e.  $FP(k)$  vs  $TP(k)$ ) is a step function as  $FP(k)$  increases by  $\frac{1}{(|S| - |C(G)|)}$  if  $G_k$  is a false positive and  $TP(k)$  increases its value by  $\frac{1}{(|C(G)| - 1)}$  if  $G_k$  is a true positive. Consequently the area under the ROC curve is given by  $\sum_k \frac{1}{(|S| - |C(G)|)} TP(k) I(G_k \in S \setminus C_G)$ . Substituting Eq. (A.2) into this equation yields Eq. (A.1) showing the equivalence of  $P(G)$  and the area under the ROC curve of  $kNN(G)$ .

### H.1 Area under precision recall curve

Given the relation of  $\bar{P}$  to the area under ROC one can also define a performance measure analogous to  $\bar{P}$  based on the area under precision-recall curve of  $kNN(G)$ . More specifically, we define  $\overline{PR} = \frac{1}{|S|} \sum_G PR(kNN(G))$ , where  $PR(kNN(G))$  is the area under the precision recall curve of  $kNN(G)$ . Although  $\overline{PR}$  produces results that are consistent with  $\bar{P}$  the area under the precision recall curve lacks the clear statistical interpretation of  $\bar{P}$ , especially in the multi-class setting [Hand and Till, 2001], hence we choose  $\bar{P}$  as our default measure of performance.  $\overline{PR}$  scores are given Table A.6. Note that *NetEmd* measures achieve the highest  $\overline{PR}$  score on all data sets.

	$RG_1$	$RG_2$	$RG_3$	RWN	Onnela et al.
$NetEmd_{G3}$	$0.962 \pm 0.024$	0.926	0.792	0.741	0.687
$NetEmd_{G4}$	$0.984 \pm 0.015$	0.951	0.810	<b>0.752</b>	<b>0.728</b>
$NetEmd_{G5}$	<b><math>0.988 \pm 0.011</math></b>	0.957	0.784	0.744	0.714
$NetEmd_S$	$0.980 \pm 0.017$	<b>0.980</b>	<b>0.908</b>	0.689	0.646
$NetEmd_{E4}$	$0.973 \pm 0.018$	0.945	0.814	0.699	0.668
$NetEmd_{DD}$	$0.830 \pm 0.066$	0.748	0.606	0.598	0.578
$Netdis_{ER}$	$0.946 \pm 0.032$	0.709	0.492	0.610	0.546
$Netdis_{SF}$	$0.965 \pm 0.021$	0.792	0.541	0.571	0.507
$GCD11$	$0.976 \pm 0.022$	0.923	0.692	0.686	0.570
$GCD73$	$0.986 \pm 0.016$	0.935	0.673	0.711	0.618
$GGDA$	$0.908 \pm 0.092$	0.877	0.618	0.511	0.566

Table A.6:  $\overline{PR}$  scores for measures and data sets considered in the main text.  $NetEmd$  measures have the highest  $\overline{PR}$  score (given in bold) on all data sets. For  $RG_1$  we calculated the value of the  $\overline{PR}$  score for each of the 16 sub-data sets. The table shows the average and standard deviation of the  $\overline{PR}$  values obtained over these 16 sub-data sets.

### I. Performance evaluation via area under precision recall curve by Yaveroglu et al.?

The area under precision recall curve (AUPRC) was used as a performance metric for network comparison measures by Yaveroglu et al. [Yaveroglu et al., 2014]. The AUPRC is based on a classifier that for a given distance threshold  $\varepsilon$  classifies pairs of networks to be similar whenever  $d(G, G') < \varepsilon$ .

A pair satisfying  $d(G, G') < \varepsilon$  is taken to be a true positive whenever  $G$  and  $G'$  are from the same class. The AUPRC is then defined to be the area under the precision recall curve obtained by varying  $\varepsilon$  in small increments. However, AUPRC is problematic, especially in settings where one has more than two classes and when classes are separated at different scales.

Figure A.7 gives three examples of metrics for a problem that has three classes: a) shows a metric  $d_1$  (AUPRC=0.847) that clearly separates the 3-classes which, however, has a lower AUPRC than the metrics given in b) (AUPRC=0.902) which confuses half of Class-1 with Class-2 and c) (AUPRC=0.896) which shows 2 rather than 3 classes. The colour scale in the figure represents the magnitude of a comparison between a pair of individuals according to the corresponding metric.

Some of the problems of AUPRC are the following. First, AUPRC is based on a classifier that identifies pairs of similar networks and hence is only indirectly related to the problem of separating classes. Moreover, the classifier uses a single global threshold  $\varepsilon$  for all networks and classes, and hence implicitly assumes that all classes are separated on the same scale. The AUPRC further lacks a clear statistical interpretation, which complicates its use especially when one has multiple classes and when precision recall curves of different measures intersect.

Despite its problems we give AUPRC values for all measures we considered in the main text in Table A.7 for the sake of completeness. Note that  $NetEmd$  measures achieve the highest AUPRC on all data sets.

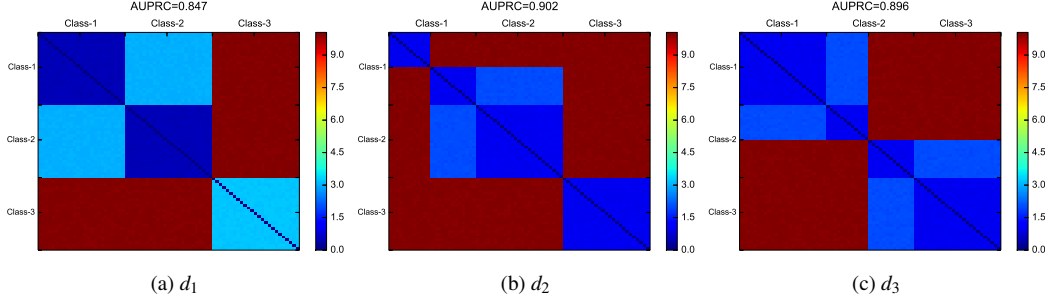


FIG. A.7: Heat maps of three measures for in an example of 3 equally sized classes. a) Metric  $d_1$  shows clear separation between the 3 classes. b)  $d_2$  shows 3 classes with half of Class-1 positioned closer to Class-2. c)  $d_3$  identifies 2 rather than 3 classes. Note that  $d_1$  has lower AUPRC than  $d_2$  and  $d_3$  despite being best at identifying the 3 classes whereas  $\bar{P}$  values for the metrics are  $\bar{P}(d_1)=1.0$ ,  $\bar{P}(d_2)=0.887$  and  $\bar{P}(d_3)=0.869$ . Similarly,  $\bar{P}\bar{R}$  values for the metrics are  $\bar{P}\bar{R}(d_1)=1.0$ ,  $\bar{P}\bar{R}(d_2)=0.893$  and  $\bar{P}\bar{R}(d_3)=0.882$ .

	$RG_1$	$RG_2$	$RG_3$	RWN	Onnela et al.
$NetEmd_{G3}$	$0.917 \pm 0.039$	0.869	0.702	<b>0.800</b>	0.756
$NetEmd_{G4}$	$0.959 \pm 0.030$	0.930	0.759	0.774	<b>0.786</b>
$NetEmd_{G5}$	<b><math>0.981 \pm 0.018</math></b>	0.957	0.766	0.722	0.757
$NetEmd_S$	$0.967 \pm 0.015$	<b>0.958</b>	<b>0.833</b>	0.702	0.672
$NetEmd_{E4}$	$0.966 \pm 0.030$	0.945	0.801	0.777	0.739
$NetEmd_{DD}$	$0.756 \pm 0.044$	0.708	0.516	0.655	0.612
$Netdis_{ER}$	$0.867 \pm 0.044$	0.579	0.396	0.607	0.621
$Netdis_{SF}$	$0.852 \pm 0.028$	0.657	0.437	0.522	0.592
$GCD11$	$0.888 \pm 0.084$	0.709	0.478	0.713	0.693
$GCD73$	$0.966 \pm 0.052$	0.858	0.571	0.736	0.743
$GGDA$	$0.815 \pm 0.176$	0.740	0.481	0.500	0.625

Table A.7: AUPRC scores for measures and data sets considered in the main text.  $NetEmd$  measures have the highest AUPRC score (given in bold) on all data sets. For  $RG_1$  we calculated the value of the AUPRC score for each of the 16 sub-data sets. The table shows the average and standard deviation of the AUPRC values obtained over these 16 sub-data sets.

## References

- M. E. J. Newman. *Networks: an introduction*. Oxford University Press, 2010.
- R. C. Wilson and P. Zhu. A study of graph spectra for comparing graphs and trees. *Pattern Recognition*, 41(9):2833–2841, 2008.
- B. Neyshabur, A. Khadem, S. Hashemifar, and S. S. Arab. Netal: a new graph- based method for global alignment of protein–protein interaction networks. *Bioinformatics*, 27:1654–1662, 2013.

- W. Ali, T. Rito, G. Reinert, F. Sun, and C. M. Deane. Alignment-free protein interaction network comparison. *Bioinformatics*, 30(17):i430–i437, 2014.
- Ö. N. Yaveroglu, N. Malod-Dognin, D. Davis, Z. Levnajic, V. Janjic, R. Karapandza, A. Stojmirovic, and N. Przulj. Revealing the hidden language of complex networks. *Scientific Reports*, 4, 04 2014.
- R. Singh, J. Xu, and B. Berger. Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences*, 105(35):12763–12768, 2008.
- O. Kuchaiev and N. Przulj. Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics*, 27(10):1390–1396, 2011.
- N. Mamano and W. B. Hayes. Sana: simulated annealing far outperforms many other search algorithms for biological network alignment. *Bioinformatics*, 33(14):2156–2164, 2017.
- S. Hashemifar and J. Xu. HubAlign: an accurate and efficient method for global alignment of protein–protein interaction networks. *Bioinformatics*, 30(17):i438–i444, 2014.
- N. Przulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183, 2007.
- T. Rito, Z. Wang, C. M. Deane, and G. Reinert. How threshold behaviour affects the use of subgraphs for network comparison. *Bioinformatics*, 26(18):i611–i617, 2010.
- G. Kossinets and D. J. Watts. Empirical analysis of an evolving social network. *Science*, 311(5757):88–90, 2006.
- K. M. Borgwardt, H.-P. Kriegel, S. V. N. Vishwanathan, and N. N. Schraudolph. Graph kernels for disease outcome prediction from protein-protein interaction networks. In *Pacific symposium on bio-computing*, volume 12, pages 4–15, 2007.
- N. Wale, I. A. Watson, and G. Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowl. Inf. Syst.*, 14(3):347–375, 2008.
- A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.
- R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.
- A. Masoudi-Nejad, F. Schreiber, Z. Kashani, and M. Razaghi. Building blocks of biological networks: a review on major network motif discovery algorithms. *IET systems biology*, 6(5):164–174, 2012.
- R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon. Superfamilies of evolved and designed networks. *Science*, 303(5663):1538–1542, 2004.
- A. E. Wegner. Subgraph covers: An information-theoretic approach to motif analysis in networks. *Phys. Rev. X*, 4:041026, 2014.
- S. Holmes, G. Reinert, et al. Stein’s method for the bootstrap. In *Stein’s Method*, pages 93–132. Institute of Mathematical Statistics, 2004.

- S. Bhattacharyya, P. J. Bickel, et al. Subsampling bootstrap of count features of networks. *The Annals of Statistics*, 43(6):2384–2411, 2015.
- W. Ali, A. E. Wegner, R. E. Gaunt, C. M. Deane, and G. Reinert. Comparison of large networks with sub-sampling strategies. *Scientific Reports*, 6:28955, 2016.
- Y. Runber, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *IEEE I. Conf. Comp. Vis.*, pages 59–66, 1998.
- A. Vázquez, A. Flammini, A. Maritan, and A. Vespignani. Modeling of protein interaction networks. *Complexus*, 1(1):38–44, 2003.
- P. Erdős and A. Rényi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61, 1960.
- I. Ispolatov, P. L. Krapivsky, and A. Yuryev. Duplication-divergence model of protein interaction network. *Phys. Rev. E*, 71(6):061911, 2005.
- D. J. Higham, M. Rašajski, and N. Pržulj. Fitting a geometric graph to a protein–protein interaction network. *Bioinformatics*, 24(8):1093–1099, 2008.
- M. Penrose. *Random Geometric Graphs*. Oxford University Press, 2003.
- M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Struct. Algor.*, 6(2-3):161–180, 1995.
- D. J. Watts and S. H. Strogatz. Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684):440–442, 1998.
- J.-P. Onnela, D. J. Fenn, S. Reid, M. A. Porter, P. J. Mucha, M. D. Fricker, and N. S. Jones. Taxonomies of networks from community structure. *Phys. Rev. E*, 86(3):036104, 2012.
- J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 177–187. ACM, 2005.
- R. C. Feenstra, R. E. Lipsey, H. Deng, A. C. Ma, and H. Mo. World trade flows: 1962-2000. Technical report, National Bureau of Economic Research, 2005.
- United-Nations-Statistics-Division. United nations commodity trade statistics database (un comtrade). <http://comtrade.un.org/>, 2015.
- P. G. Maugis, C. E. Priebe, S. C. Olhede, and P. J. Wolfe. Statistical inference for network samples using subgraph counts. *ArXiv e-prints*, 2017.
- E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. *Adv. Neur. In.*, 15:505–512, 2003.
- B. Mohar, Y. Alavi, G. Chartrand, and O. R. Oellermann. The laplacian spectrum of graphs. *Graph theory, combinatorics, and applications*, 2(871-898):12, 1991.
- F. R. K. Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

- A. Banerjee and J. Jost. On the spectrum of the normalized graph laplacian. *Linear algebra and its applications*, 428(11-12):3015–3022, 2008.
- J. Gu, J. Jost, S. Liu, and P. F. Stadler. Spectral classes of regular, random, and empirical graphs. *Linear algebra and its applications*, 489:30–49, 2016.
- P. Yanardag and S. V. N. Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1365–1374. ACM, 2015.
- C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- N. Shervashidze, S. V. N. Vishwanathan, T. Petri, K. Mehlhorn, and K. M. Borgwardt. Efficient graphlet kernels for large graph comparison. In *AISTATS*, volume 5, pages 488–495, 2009.
- I. Barnett, N. Malik, M. L. Kuijjer, P. J. Mucha, and J.-P. Onnela. Feature-based classification of networks. *CoRR*, abs/1610.05868, 2016.
- Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International Conference on Machine Learning*, pages 2014–2023, 2016.
- N. Shervashidze, P. Schweitzer, E. J. v. Leeuwen, K. Mehlhorn, and K. M. Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(Sep):2539–2561, 2011.
- T. Hočevár and J. Demšar. A combinatorial approach to graphlet counting. *Bioinformatics*, 30(4):559–565, 2014.
- R. P. Brent. An algorithm with guaranteed convergence for finding a zero of a function. *Comput. J.*, 14(4):422–425, 1971.
- M. Thüne. *Eigenvalues of matrices and graphs*. PhD thesis, University of Leipzig, 2013.
- A. K. Debnath, R. L. Lopez de Compadre, G. Debnath, A. J. Shusterman, and C. Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *J. Med. Chem.*, 34(2):786–797, 1991.
- K. M. Borgwardt, C. S. Ong, S. Schonauer, SVN Vishwanathan, A. J. Smola, and H.-P. Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl\_1):i47–i56, 2005.
- Mahito Sugiyama and Karsten Borgwardt. Halting in random walk kernels. In *Advances in neural information processing systems*, pages 1639–1647, 2015.
- P. D. Dobson and A. J. Doig. Distinguishing enzyme structures from non-enzymes without alignments. *J. Mol. Biol.*, 330(4):771–783, 2003.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.
- C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

- S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi. Kernel methods on riemannian manifolds with gaussian rbf kernels. *IEEE transactions on pattern analysis and machine intelligence*, 37(12):2464–2477, 2015.
- R. Luss and A. d’Aspremont. Support vector machine classification with indefinite kernels. In *Advances in Neural Information Processing Systems*, pages 953–960, 2008.
- B. Haasdonk. Feature space interpretation of svms with indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):482–492, 2005.
- E. N. Gilbert. Random plane networks. *J. Soc. Ind. Appl. Math.*, 9(4):533–543, 1961.
- A. L. Traud, P. J. Mucha, and M. A. Porter. Social structure of facebook networks. *Physica A*, 391(16):4165–4180, 2012.
- H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.
- C. Stark, B.-J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers. Biogrid: a general repository for interaction datasets. *Nucleic Acids Res.*, 34(suppl 1):D535–D539, 2006.
- J. Das and H. Yu. Hint: High-quality protein interactomes and their applications in understanding human disease. *BMC Syst Biol.*, 6(1):92, 2012.
- S. V. Rajagopala, P. Sikorski, A. Kumar, R. Mosca, J. Vlasblom, R. Arnold, J. Franca-Koh, S. B. Pakala, S. Phanse, A. Ceol, et al. The binary protein-protein interaction landscape of escherichia coli. *Nat. Biotechnol.*, 32(3):285–290, 2014.
- L. Jure and A. Krevl. SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data>, 2014.
- D. J. Hand and R. J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45(2):171–186, 2001.