Generative modelling and inverse problem solving for networks in hyperbolic space

Alessandro Muscoloni

DISSERTATION

to achieve the academic degree

Doktoringenieur (Dr.–Ing.)

5th July 2019

Faculty of Computer Science

Technische Universität Dresden

Committee:

Prof. Thorsten Stufe (Chair)

Dr. Carlo Vittorio Cannistraci (Subject expert)

Prof. Michael Schroeder (1. Reviewer)

Prof. Giuseppe Mangioni (2. Reviewer)

Prof. Gerhard Weber (Committee member)

Acknowledgements

After having reached this scientific and career achievement, I would like to personally and warmly thank all the people that contributed to make this possible.

To my group leader Carlo, for having guided and supported me through these three years of continuous growth, as well as for sharing my same exploratory mentality during our travels.

To my academic supervisor Michael, for the wise suggestions during our meetings, in particular towards the preparation of the dissertation.

To the secretaries Gloria, Michelle, Janett and Claudia, for the patient support on the boring bureaucracy.

To all my colleagues, in particular to Claudio, Aldo and Alberto for the wonderful time spent together also outside work, as well as to Sara, Phine, Ali, Umberto, Ilyes and all the people that helped to create a positive working environment.

To my girlfriend Ceciel, for the lovely experiences shared in the last years.

To my family, for always being close despite the distance.

Papers & Contributions

The three papers of the publication-oriented dissertation are listed below. The detailed contributions are reported for each paper.

- **Paper A**

A. Muscoloni, J. M. Thomas, S. Ciucci, G. Bianconi, C. V. Cannistraci. (2017). Machine learning meets complex networks via coalescent embedding in the hyperbolic space. *Nature Communications*, 8, 1615.

Alessandro Muscoloni's contributions:

- Design and implementation of the code for the coalescent embedding algorithm and for the evaluation measures HD-corr, C-score and GR-score.
- Computational simulations for HD-corr, C-score and GR-score evaluations, community detection and rich-club analyses.
- Analysis and interpretation of the results.
- Realization of figures and tables.
- Analysis of the computational complexity of the coalescent embedding variants.
- Drafting of the article.

- **Paper B**

A. Muscoloni, C. V. Cannistraci. (2018). A nonuniform popularity-similarity optimization (nPSO) model to efficiently generate realistic complex networks with communities. *New Journal of Physics*, 20, 052002.

Alessandro Muscoloni's contributions:

- Design of the nonuniform distributions for the nPSO model.
- Design and implementation of the code for the nPSO model.
- Computational simulations.
- Analysis and interpretation of the results.
- Realization of figures and tables.
- Mathematical demonstration of the equivalence of the implementations for link generation.
- Mathematical analysis of the computational complexity.
- Drafting of the article.

- **Paper C**

A. Muscoloni, C. V. Cannistraci. (2018). Leveraging the nonuniform PSO network model as a benchmark for performance evaluation in community detection and link prediction. *New Journal of Physics*, 20, 063022.

Alessandro Muscoloni's contributions:

- Implementation of the code.
- Computational simulations.
- Analysis and interpretation of the results.
- Realization of figures and tables.
- Drafting of the article.

Table of Contents

Abstract

The investigation of the latent geometrical space behind complex network topologies is a fervid topic in current network science and the hyperbolic space is one of the most studied, because it seems associated to the structural organization of many real complex systems. The popularity-similarity-optimization (PSO) generative model is able to grow random geometric graphs in the hyperbolic space with realistic properties such as clustering, small-worldness, scale-freeness and rich-clubness. However, it misses to reproduce an important feature of real complex systems, which is the community organization. Here, we introduce the nonuniform PSO (nPSO) generative model, a generalization of the PSO model with a tailored community structure, and we provide an efficient algorithmic implementation with a O(EN) time complexity, where N is the number of nodes and E the number of edges. Meanwhile, in recent years, the inverse problem has also gained increasing attention: given a network topology, how to provide an accurate mapping into its latent geometrical space. Unlike previous attempts based on a computationally expensive maximum likelihood optimization (whose time complexity is between O(N^3) and O(N^4)), here we show that a class of methods based on nonlinear dimensionality reduction can solve the problem with higher precision and reducing the time complexity to O(N^2).

Part I: Research Summary

1. Introduction

The Oxford English Dictionary (OED, 2019) provides several definitions for the term *network*, among which the most general and intuitive is 'collection of interrelated things'. Network science is the academic interdisciplinary field that, combining theories from mathematics, physics, computer science, statistics and more, aims at studying physical systems represented as networks. In particular, the elements of the system are represented by *nodes* (or *vertices*) and the interactions between them are represented by *links* (or *edges*). The way in which the connections between the elements are arranged defines the *structure* (or *topology*) of the network. Most of the real networks intended to be modeled are considered complex networks, meaning that they exhibit structural features that are not purely regular nor purely random [1]. From the historical point of view, the beginning of network science dates back to 1736, when Leonhard Euler published the famous work on the Seven Bridges of Königsberg [2]. The problem described four land areas separated by a river and connected by seven bridges, asking whether it was possible to visit all the four land areas crossing each bridge once and only once. Euler noticed the only information relevant for approaching the problem was the interconnectivity between the land areas through the bridges, and therefore he could reformulate the problem in abstract terms representing the land areas as vertices and the bridges as edges. This corresponded to the foundation of graph theory, which studies *graphs* as mathematical structures used to model pairwise relations between objects, being therefore the mathematical framework for network science.

In the scientific literature the terms network, node and link from network science are often used interchangeably with the terms graph, vertex and edge from graph theory. However, for the sake of clarity, networks and the related terminology refer to the physical systems that are represented, whereas we use graphs while referring to the mathematical representation of these systems. The mathematical object usually adopted to describe a graph is the *adjacency matrix*, which is in the simplest case a binary square matrix having as many rows and columns as the number of nodes, and where each element in position (i,j) indicates whether the nodes i-th and j-th are connected (value 1) or not (value 0) [1].

Being the definition of network very general, it can be adopted to describe systems of completely different nature and scale. For example, in a protein-protein interaction network, the nodes are protein molecules and the links represent physical contacts between them. At a larger scale, a typical example is a social network, modeling social interactions between people, like friendship or collaboration. Countless further examples could be mentioned, like brain networks, ecological networks, economical networks, technological networks, telecommunication networks and so on [1].

Networks can also be classified depending on different ways in which pairs of nodes can be connected. In fact, links might have a directionality, beginning from a given node and ending to another node. In this case the network is known as *directed network* and an example is given by the World Wide Web, where the links between pages are directed. In the opposite scenario, when the directionality is meaningless, we would refer to an *undirected network*, as in a Facebook social network where the links represent the friendship between two users, which is a mutual relationship. A network might also allow a link between a node and itself, known as *self-loop*, for example to describe autocatalytic reactions in metabolic networks. Finally, we can have a *weighted network*, where a weight is assigned to each link, for example to indicate the strength of a physical interaction between molecules, the distance between two geographical points in a road map or the average amount of passengers in a flight between two cities [1].

Structural properties

Provided a preliminary overview on networks and the basic terminology, we will now introduce the main topological properties commonly adopted to characterize the structure of a network. The two simplest features consist of the *number of nodes* (*N*) and the *number of links* (*L*), which are both indicators of the network size. Their values might differ widely between systems of different scale, going for example from a small friendship network within a sport club with tens of athletes, up to a large online social network with millions of users and billions of interactions between them. Another property related to the number of nodes and links is the network *density*, which is defined as the ratio between the number of links over the number of possible links. Many real networks are *sparse*, meaning that the number of links is much smaller than the number of possible links (low density) [3]. A key feature of each node of the network is its *degree*, which is the number of links it has to other nodes. For example in a social network modelling the friendship within a community, the degree of each individual indicates the number of friends. A network property closely related to the density is the *average degree*, simply defined as the average of the node degrees [3]. The *degree distribution*, which is the probability distribution for a certain node having a certain degree, has a central role in characterizing the network organization and will be discussed more in details in later sections. An important concept in networks is the one of *path length*. A *path* is a sequence of links visiting a sequence of nodes, which usually have to be all distinct. Its length is simply the number of links in the sequence. In networks, the distance between two nodes is often represented by the *shortest path* between them, which is the path with the lowest amount of links. Shortest paths are at the base of many topological properties, for example the *diameter*, defined as the longest shortest path, or the *characteristic path length*, defined as the average of the shortest path between all pairs of nodes [4].

Another fundamental concept used to characterize the structure of a network is *clustering*. For a given node, its *local clustering coefficient* is the ratio between the number of links between its neighbors over the number of possible links that its neighbors could have. In other words, it is equal to the density of the subgraph of its neighbors and it is therefore a measure of local density. The *average clustering coefficient* of a network is simply defined as the average of the local clustering coefficient over all the nodes [4]. More intuitively, in a network with a high clustering the nodes tend to cluster together by closing triangles. For example, social networks usually exhibit a high clustering since if the individual A is friend with B and B is friend with C, then it is also likely that A is friend with C. Indeed, the concept of clustering in a network is strictly related to the concept of *transitivity* [5].

Looking for patterns of topological organization in networks, many real systems present a *community structure*. In intuitive terms, the network is composed of several communities (groups of nodes), such that nodes that belong to a community have a higher probability to link to the other members of that community than to nodes that do not belong to the same community [6]. These groups of densely connected nodes can have completely different meanings depending on the system in observation. In a social network, a community can represent a tight group of friends, people sharing the same hobbies or the same job. In a protein-protein interaction network it corresponds to proteins with a similar functionality, whereas in a citation network it represents a set of scientific papers with a related topic. Many algorithms have been developed for detecting communities in networks and also a measure, the *modularity*, indicating the extent to which a network can be partitioned into communities [7].

Network models

One of the main goals of network science is to develop models that are able to reproduce features as close as possible to the ones of real networks. A large amount of models have been currently designed, each of them trying to emulate certain network characteristics [3]. In this section we will recall three models that certainly marked the history of network science.

The first well known model for generating *random networks* is the Erdős-Rényi model introduced in 1959 [8]. As presented in the original publication, the model has two parameters, the number of nodes N and the number of links L, and the network is generated by choosing at random (with equal probabilities) L links among all the possible pairs of nodes. In a slight variation introduced by Gilbert, the model has as parameters the number of nodes N and the connection probability p, and for each pair of nodes a link is established with probability p [9]. Networks generated with the Erdős-Rényi model exhibit a small characteristic path length and a small average clustering coefficient. The degree distribution follows a binomial distribution, which becomes a Poisson distribution for large values of N [3].

In 1967 Stanley Milgram published the results of experiments investigating "the small-world problem" [10]. In the experiment, several random people living in the center of the USA were asked to send a letter to a target person living in Massachussets, or, in case they did not know personally the target person, they should have sent the letter to someone they knew and that could know the target person. By following the trajectory of the letters, the resulting network showed two main properties: the average path length was low (in particular around 6, from which the popular expression "six degrees of separation"), and the local clustering coefficient was high. While the first feature is found in the Erdős-Rényi model, the second one is lacking.

In 1998, Watts and Strogatz proposed a new network model able to reproduce both the properties [4]. The parameters required are the number of nodes N, the average degree k and the rewiring probability p. Starting from a ring lattice, placing the nodes in a circle and connecting each node with the k nearest neighbors, each link is then rewired with probability p. In the extreme case $p = 1$, the generated network is an Erdős-Rényi network, with low characteristic path length and low average clustering coefficient. In the opposite case $p = 0$, the network remains a ring lattice, with high characteristic path length and high average clustering coefficient. However, there is a region of p values for which the characteristic path length is low and the average clustering coefficient is high, typical of *small-world networks*. The degree distribution, like for the Erdős-Rényi model, is a Poisson distribution [3].

In 1999, Barabasi and Albert argued that many large real networks have a degree distribution following a power-law function, which arises naturally introducing two generic mechanisms: growth and preferential attachment [11]. In other words, the network expands by the addition of new nodes, and these new nodes preferentially attach to nodes that are already well connected. According to the generative procedure of the model, starting from a small number of nodes, at every step a new node is added and it connects to *m* nodes already present in the network, where the links are chosen with probabilities proportional to the degrees of the nodes. The resulting networks with power-law degree distribution are also reported as *scale-free networks* [11]. The main characteristic that makes the power-law different from the Poisson degree distribution of the previous models is the fat tail, implying that there are many nodes with few connections and few nodes with many connections, and such rich nodes are known as *hubs*. However, the rareness or universality of scale-free networks, as well as the definition of scale-free network itself, it is still a controversial topic [12]–[14].

2. Related work

Given a general introduction on network science, this section will focus more on the specific subfield on which the dissertation is based, which is *network geometry*. It will summarize, mainly in chronological order, the most important papers that provided significant contributions to the rapid growth and interest that this topic has gained around the last decade. Both the models and the inference methods that represent the milestones in this subfield will be reported, together with examples of applications from the literature. This overview will give the knowledge required in order to present, in the next section, the results of the three papers that characterize this dissertation. Note that some models will be commented in this section in a more qualitative and conceptual fashion, without going into the mathematical details, since they are already reported within the papers.

In 2008 Serrano et al. introduced the concept of *hidden metric space* [15]. The basic idea of hidden metric space of a complex network is that the nodes are located at certain geometrical coordinates, such that for each pair of nodes a distance is defined, and nodes that are closer to each other in the space are more likely to be connected in the network topology [15]. The original motivation of Serrano et al. was that self-similarity and scale-freeness, traditionally known as characteristics of certain geometric objects such as fractals, are still not properly defined in a geometric sense for complex networks, since many complex networks are not explicitly embedded in a physical space [15]. In order to test whether real networks could exhibit self-similarity, they designed a degree-normalization procedure, considering increasing degrees as thresholds, and analyzing for each threshold the subgraph consisting of the nodes with degree higher than the threshold. After analyzing few real networks, they concluded that the topological characteristics of the subgraphs, like degree distribution and clustering coefficient, were very similar, and therefore those networks could be considered self-similar with respect to the renormalization procedure [15]. As a further validation, they noticed that the self-similarity was not present after a randomization of the original topology while preserving the degree distribution (in particular, in this case the clustering was not self-similar). In addition to the self-similarity, another characteristic suggesting the presence of a hidden metric space behind complex networks was the high clustering measured in the real topologies [15]. Indeed, if the distance between the nodes in the hidden metric space satisfies the triangle inequality, then a high concentration of triangles, and therefore high clustering, will arise naturally. The triangle inequality states that for any triangle, the sum of the lengths of any two sides must be greater than or equal to the length of the remaining side. In intuitive words, considering that nodes that are closer are more likely to be connected, if A is close (and connected) to B and B is close (and connected) to C, then A cannot be too far away from C, therefore they will likely be connected too, closing the triangle [15]. In the rest of their work, Serrano et al. introduced a network model where the nodes exist in a one-dimensional metric space (a circle) underlying the network topology, and that is able to generate networks with a scale-free degree distribution, strong clustering and the small-world property. They also emphasized that there were already in literature models of network embedded in a metric space, but that none of them could simultaneously reproduce all the characteristics mentioned above [15].

In 2009 Boguña et al. presented a second advantage of the adoption of a hidden metric space [16]. Not only it can offer an explanation for the topological characteristics typically observed in real networks, but it can also provide a plausible framework to justify the high *navigability* efficiency that networks often exhibit [16]. Many real systems have transport as their primary function, such as Internet or airport networks. When the elements of the system have a full view of the global network topology, finding the shortest route from the source to the target destination has a well-known computational solution. However, when the elements of the system do not have any global knowledge, for example in signaling pathways and neural networks, the protocol followed for reaching a highly efficient routing is not always clear. Boguña et al. adopted a greedy routing protocol to investigate this second case, in which the nodes of the network need to send information to a target destination only exploiting local information. In particular, each node knows only the position of its neighbors and of the destination in the hidden metric space, and it forwards the information to its neighbor closest to the destination [16]. They generated synthetic networks adopting the model of Serrano et al. [15] in order to reproduce characteristics observed in real networks and also to have a hidden geometric space behind the network topology. After several experiments they noticed that within the hidden metric framework and greedy routing protocol adopted the networks could route information with high efficiency, although the elements of the system were lacking global knowledge [16]. The main motivation that inspired the work of Boguña et al. was the scalability problem of the Internet architecture. In order to route packets to target destinations, the routers need to communicate to keep a coherent view of the global Internet topology. Due to the continuous growth in size and dynamics of the Internet, this communication processing overhead might become a crucial bottleneck for routing scalability in the near future. The adoption of a hidden metric space for the Internet network would theoretically remove the need for maintaining global knowledge allowing the routers to greedily forward packets to the target destination based on the hidden distances [16].

In 2010 Krioukov et al. [17] make a further step in the field of network geometry. Under the assumption that there is a *hyperbolic geometry* underlying the topology of complex networks, then strong clustering and a power-law degree distribution emerge naturally. On the other side, assuming that the network has a hidden metric space and that the degree distribution is power-law, then the metric distances can be rescaled such that the hidden geometry is hyperbolic. Considering the greedy routing framework previously discussed by Boguña et al. [16], they also show that the routing efficiency is maximized if the hidden metric space underlying the network topology is hyperbolic. Furthermore, in order to test the robustness of the routing in case of damages to the Internet topology, they simulated the failure of certain percentages of the network links, highlighting a high robustness of the routing efficiency in presence of such perturbations events [17]. These results of this work paved the way for a large variety of studies and investigations about the hidden hyperbolic geometry of complex networks, which is currently a very hot topic in the area.

In line with the motivation behind the work of Boguña et al. [16] and the discovery of Krioukov et al. [17], in 2010 Boguña et al. introduced a method to map the Internet network to a hyperbolic space [18]. The paper shows that using this map and the greedy routing protocol, the scaling properties of the Internet routing become theoretically close to the best possible, potentially solving the problem of sustaining the Internet scalability [18]. The mapping method adopted is not specific for the Internet network, it is generic and can be applied to any other complex network with hidden metric structure and power-law degree distribution. The mapping method is based on likelihood maximization, therefore it aims at maximizing the likelihood that the observed Internet topology has been generated by the network model [18]. In particular, the model considered is the one described by Krioukov et al. [19], which is similar to the one previously introduced by Serrano et al. [15]. As indicated by the authors, methods based on maximum likelihood estimation can be very time consuming and the running times might become prohibitive for substantially large networks [18].

In 2012, Papadopoulos et al. [20] introduced a new fundamental concept for describing the mechanisms characterizing the growth of complex networks. Since the well-known paper of Barabasi et al. [11], it has been widely accepted that the *popularity* was a main attractive force during the network evolution, where new nodes entering the system preferentially link to nodes already popular in the network. This "rich get richer" process was naturally giving rise to a degree distribution following the power-law function [11]. In this work, Papadopoulos et al.

[20] show that popularity is only one dimension of attractiveness, with another dimension being the *similarity*. Indeed, nodes that are more similar have a higher chance of establishing a connection, regardless of the fact that they are popular or not [20]. In the social science this effect is known as homophily, and it has been observed in many other real networks [21]. Therefore, Papadopoulos et al. [20] developed a new framework in which new connections optimize a certain trade-off between the popularity and the similarity between the nodes. The simplest way to model the popularity would be the node birth time t. If there are no other factors in play, the older nodes have higher likelihood to become popular and attract more connections. In order to model the similarity, the nodes are randomly placed with an angular coordinate ϑ on a circle, which represents a simple similarity space where nodes at lower angular distance are more similar. In this basic model, the network is initially empty, and at each subsequent time step one new node enter in the network, linking to a prefixed number m of the existing nodes, chosen as the ones that minimize the product between birth time (popularity) and angular distance (similarity) [20]. Papadopoulos et al. [20] show that this basic model has an interesting geometrical interpretation. If we map the nodes not on a circle but on a plane adding also a radial coordinate, defined as the logarithm of the birth time $r = \ln t$, then establishing the links optimizing the product between birth time and angular distance is equivalent to connecting to the nodes at the lowest *hyperbolic distance*. Assuming the network with a hidden hyperbolic space, the hyperbolic distance can therefore be adopted as a unique metric representing the trade-off between the two forces of attraction in the network growth: radial popularity and angular similarity [20]. Papadopoulos et al. [20] comment also that there are crucial differences between the pure preferential attachment of the Barabasi-Albert model [11] and the model here proposed. The most important regards the clustering coefficient, which becomes asymptotically zero for increasing network size in the Barabasi-Albert model, whereas this model is able to reproduce the strongest possible clustering for a given degree distribution [20]. To the basic model, initially described for a more conceptual discussion, Papadopoulos et al. [20] added two main modifications in order to provide the possibility of tuning the clustering coefficient and the exponent of the power-law degree distribution. The first modification consists in introducing the concept of temperature, which intuitively indicates whether the network topology is congruous or not with the underlying metric space. At zero temperature the topology maximally reflects the underlying geometry, the new nodes connect only to the geometrically closest existing nodes and the clustering coefficient is maximized. At increasing temperature, new nodes establish links with a connection probability inversely related to the geometrical distance, therefore also connections between nodes far from

each other are possible, and the clustering coefficient decreases [20]. The second modification consists in introducing the concept of popularity fading, according to which, while the network grows, the existing nodes increase their radial coordinate drifting away from the center of the hyperbolic disk, which is the point of maximum popularity. Changing the speed of this popularity fading allows to tune the exponent of the power-law degree distribution. Papadopoulos et al. [20] let also notice that, by letting the different nodes move with different speeds, this geometric optimization model can be mapped to the Bianconi-Barabasi model [22]. This model is an extension of the Barabasi-Albert model [11] adding to the concepts of growth and preferential attachment the one of fitness. Each node has a value of fitness that indicates its ability to attract new links regardless of its degree. The introduction of the fitness model allows also to new nodes entering the system to become popular [22]. Note that more details on the popularity-similarity optimization (PSO) models can be found in the Methods sections of the Papers A, B and C of the dissertation.

While the PSO model [20] lead to a significant progress in understanding the role of the hidden metric space behind the structure and dynamics of real complex networks, in order to translate this knowledge into practical applications a reliable network mapping method is required. An embedding method, given the observed topology of the real networks, infers the geometric coordinates of the nodes in the hidden metric space. Only provided such mapping, further analysis and applications based on the hidden geometry of the network would be possible. Given the ability of the PSO model to generate synthetic networks that well resemble real systems over a wide range of structural and dynamical features, Papadopoulos et al. [23] raised the question of whether it would be possible to map a network into the hyperbolic space in a way congruent to the PSO model. In order to answer to this question, Papadopoulos et al. [23] design an embedding methods that they named HyperMap. HyperMap, like the mapping method previously introduced by Boguña et al. [18], is based on maximum likelihood estimation. The algorithm replays the geometric growth of the network according to the PSO model and assigns coordinates to the nodes by maximizing the likelihood that the network is produced by the model [23]. The authors applied HyperMap to the Autonomous Systems topology of the Internet. As a first result, they showed that the mapping highlights soft communities of Autonomous Systems belonging to the same geographic region, where by soft communities they indicate groups of nodes located close to each other in the space. This is reasonable, since Autonomous Systems belonging to the same country tend to connect more densely between each other than to the rest of the world [23]. Given the Internet embedding, Papadopoulos et al. [23] also tested link prediction as practical application, and in particular the prediction of missing links [23]. They adopted a standard framework of evaluation in which a percentage of links is randomly removed from the original topology, the remaining topology is provided in input to the algorithm, which provides in output a score for each nonobserved link, representing the likelihood of that link to exist. The performance is then evaluated by verifying whether the missing links originally removed obtained higher likelihood with respect to the other nonobserved links, computing several possible evaluation metrics, such as precision [24]. In the case of HyperMap, the hyperbolic distance was used to assign likelihood scores: the smaller the hyperbolic distance between two disconnected nodes, the higher the likelihood of the nonobserved link between them to exist [23]. The performance of HyperMap resulted similar to the one of basic link predictors adopted in literature, whereas it was better when hard-to-predict links (between low degree nodes without common neighbors) where evaluated [23]. At last, similarly to Boguña et al. [18], Papadopoulos et al. [23] applied the Internet mapping also for testing the navigability of the network, obtaining that the greedy routing is highly efficient, in agreement with previous results [23].

In the same year, Papadopoulos et al. [25] developed an extended variant of HyperMap, which adopts also common neighbors' information in order to infer the coordinates of the nodes in the hyperbolic space. In particular, while in HyperMap the likelihood function to maximize only contains information about the connections and disconnections between the nodes, in this new approach the likelihood function also includes information about the number of common neighbors between the nodes [25]. The authors state that this extension leads to a more accurate inference, although the approach is computationally expensive with a complexity of O(N^4), being *N* the number of nodes in the network. Therefore they also developed a hybrid approach adopting the likelihood function of the new variant, based on common neighbors' information, only for inferring the coordinates of high degree nodes, and the likelihood function of HyperMap for the remaining nodes, arguing that the computational complexity is reduced to O(N^3) without significant loss in mapping accuracy [25]. Note that more details on the HyperMap method and its extension (referred to as HyperMap-CN) can be found in the Methods sections of the Papers A and C of the dissertation.

In 2013, Mangioni and Lima [26] presented a separate work that also builds on the results of Krioukov et al. [17]. Analogously to Papadopoulos et al. [20], they noticed that the preliminary model presented by Krioukov et al. [19] was static, and therefore it did not properly reproduce the dynamical properties of many real networks, which in most of the cases evolve and grow over time, with new nodes entering the system. Therefore, they proposed a new model that is a variation of the one presented by Krioukov et al. [19], with the main differences that the network grows over time and that the space is discretized, meaning that the nodes can only occupy specific discrete slots within the hyperbolic space. The focus of the work of Mangioni and Lima [26] was to present a model more devoted to be adopted in practical applications such as building overlay networks in distributed computing environment [27]. The authors also commented that the discretized growing model developed conserves the desirable properties of the static model of Krioukov et al. [19], such as a power-law degree distribution and an extremely efficient greedy routing.

After the introduction of the PSO model from Papadopoulos et al. [20], in 2015 Zuev et al. [28] perform an additional step. The authors argue that, although there is no consensus in what are the most common properties of real networks, power-law degree distribution, high clustering and community structure are certainly frequent [28]. While we have already discussed how the PSO model is able to reproduce the first two features in growing networks, a simple mechanism that provides also a non-trivial community structure is still lacking [28]. In order to fill this gap, Zuev et al. [28] introduce the concept of geometric preferential attachment (GPA). The model proposed is similar to the PSO but with a new main assumption. The similarity space is not equally attractive, instead it contains hot regions that are more attractive for the new nodes entering the system [28]. For example, in a collaboration network where nodes are scientists, a hot region can represent a hot working area in science. In order to model this new hypothesis, the new nodes are not placed in the similarity space with uniform probability, but they are placed in the hot regions with higher probability, therefore generating regions of higher node density. As a consequence, the distribution of the node angular coordinates in the similarity space is not homogeneous as in the PSO model, but heterogeneous [28]. The attractiveness of the regions is characterized by two factors in the model. The first one is the number of existing nodes that lie in the proximity of that regions, intuitively meaning that regions where many nodes are present are more attractive. The second factor is a parameter of initial attractiveness, which allows mainly to tune the heterogeneity of the angular coordinates distribution [28]. The authors comment that, while in the PSO the choice of a position in the similarity space is a passive decision performed uniformly at random, in the GPA it can be considered an active decision made by the node depending on the attractiveness of the different locations. The heterogeneous distribution of the angular coordinates of the nodes encodes a non-trivial community structure, where the communities are indicated by clusters of spatially close nodes [28]. In particular, Zuev et al. define a soft community as a group of nodes separated from the rest of the network by two gaps that exceed a certain critical value, and based on this definition they also propose a method to detect the communities from the similarity space [28]. In summary, starting from the framework of hyperbolic network geometry, considering the popularity and similarity as driving forces in the network evolution, and adding the mechanism of preferential attachment of the nodes to hot geometric regions, the GPA model is able to grow networks with a power-law degree distribution, strong clustering and an emergent soft community structure [28]. Note that more details on the GPA model can be found in the Methods sections of the Paper B of the dissertation.

In 2016 Krioukov [29] brings further support to the field of network geometry, motivated by the problem that it is usually not possible to know if a given real network has a hidden geometry, or in other words, if such real network is an element in the set of random geometric graphs. In his work, Krioukov [29] demonstrates that given random graphs with expected degree and clustering of each node fixed to some constants, if the clustering if sufficiently strong, then they are equivalent to random geometric graphs on the Euclidean line. The degree distribution of such random geometric graphs is the Poisson distribution, however it can be enforced to be a power-law distribution as typically observed in real networks by enforcing a hyperbolic geometry [29].

In the same year, Kleinberg et al. [30] extend the study of hidden metric spaces to multiplex networks. Despite the presence of contradictory terminology in literature [31], in this case a *multiplex* or *multilayer network* is considered a system composed of multiple networks having all the same set of nodes but with different patterns of interactions, usually representing different types of relationships between the nodes [31]. For example, in a transportation multilayer network the nodes could be cities and different layers might represent different types of connections between the cities (bus, train, flight and so on). In this work they show that multiplex systems are not random combinations of single network layers. Instead, they are organized in specific ways dictated by hidden geometric correlations between the layers, and they are hidden since they cannot be observed looking at the individual network topologies [30]. In their analysis, each single network layer is separately mapped into the hyperbolic space (in this case using the HyperMap method), assigning to each node radial and angular coordinates for each layer. The coordinates of the nodes can then be compared across layers, and the authors found that they are significantly correlated, meaning also that the distances between the nodes in the underlying hyperbolic spaces of the each network layer are also significantly correlated [30]. As a first result, the authors discovered an overabundance of two-dimensional soft communities, defined as sets of nodes that are similar (located at small angular distance) in both layers of the multiplex. These clusters of similar nodes do not appear in reshuffled counterparts of the systems, where the node ids are randomly permutated [30]. As second result, the authors show that the strong geometric correlations between layers allow to perform an accurate trans-layer link prediction, meaning that the hyperbolic distances in one layer can be used to predict the connections in another layer [30]. Kleinberg et al. comment that this could be useful when there is knowledge of the connections between nodes in one context, for example structural connections between brain regions, and we need to perform a prediction in another related context, for example the likelihood of functional connections between the same brain regions [30]. At last, their work investigated the importance of the geometric correlations for the navigability of the networks. While it has been already shown in previous works that real networks, like the Internet network, has a highly efficient greedy routing when mapped in the hyperbolic space [18], this has not been analyzed by exploiting multiple layers yet. We recall that in the single-layer greedy routing protocol, the node send the information to their neighbor that is closest to the destination in the geometric space. In the multilayer scenario, the authors defined the process as mutual greedy routing, where the information can flow also across layers, therefore the node forwards the message to its neighbor that is closest to the destination in any of the layers of the system [30]. If we think about the previous example of multilayer transportation network, where different layers represent different types of transport, then such protocol would be equivalent for example to go from a source city to an intermediate city by train, and then to switch layer and proceed from the intermediated city to the destination city by bus. In particular, in this paper they showed that the mutual greedy routing in the multiplex network consisting of the IPv4 and IPv6 topologies is more efficient than the greedy routing performed in the single IPv4 and IPv6 networks [30]. In 2017, Allard et al. [32] made a further step towards the generalization of network models based on hyperbolic geometry and extended the current framework also to weighted networks. They introduce a new general model that can be used to quantify the level of coupling between the topology, the weights and the hidden metric space, and also to reproduce both the topology and weights of real networks. Their results suggest that the formation of links and the assignment of their weights are governed by different processes [32].

We conclude this section with a recent work released in 2019 as an arXiv preprint, where Kitsak et al. [33] performed a wide investigation about the usage of hyperbolic geometry for link prediction, due to contradictory results presented in the previous literature. We recall that link prediction with hyperbolic geometry is a two-step procedure. First, the network has to be mapped in the hyperbolic space, meaning that the coordinates of the nodes need to be inferred. Second, from the given coordinates the nonobserved links that are more likely to be missing can be identified. As a first result of the article the authors notice that some evaluation measures of link prediction accuracy are extremely sensitive to inaccuracies in the inference of the hyperbolic coordinates of the nodes. For such reason, the authors developed a new method for network embedding in the hyperbolic space that, as they report, maximizes the accuracy of such inference. As second main result, the work highlights that there are many link prediction methods able to infer more obvious and easy-to-predict links, among which hyperbolic link prediction is usually not the best but still competitive, whereas it becomes the best when the task is to predict less obvious and hard-to-predict missing links, such as missing links in largely incomplete networks and missing links between nodes without common neighbors [33].

3. Results

After the general introduction on network science and a review of the most important works related to the field of network geometry, this section will focus on the three papers of the dissertation. Fistly, the content of each paper is briefly reported, then the main results of each paper will be discussed.

In Paper A we introduce a new class of topological machine learning algorithms for embedding networks in the hyperbolic space, named ‘coalescent embedding’, which reduces the computational complexity to $O(N^2)$, versus $O(N^3)$-$O(N^4)$ of the state-of-the-art methods based on maximum likelihood approaches. We adopted the PSO model as a framework to compare the mapping accuracy and greedy routing efficiency of coalescent embedding against state-of-the-art methods, using three evaluation measures introduced in this paper: HD-correlation, C-score and GR-score. As example of application, we exploit the geometrical information inferred using coalescent embedding to boost community detection algorithms on real networks. At last, we show how the coalescent embedding algorithms are able to provide mappings not only in the two-dimensional hyperbolic disk but also in the three-dimensional hyperbolic sphere.

In Paper B we introduce the nonuniform PSO (nPSO) model, which generates synthetic networks in the hyperbolic space, where heterogeneous angular node attractiveness is forced by sampling the angular coordinates from a tailored nonuniform probability distribution. The model is able to reproduce synthetic networks with high clustering, small-worldness, scale-freeness and a desired community organization, with the possibility to fix the number of communities, their relative sizes and to tune the mixing between them. An efficient implementation of the nPSO model is provided, with computational complexity of O(EN). In addition, we investigate the detectability of the communities by a state-of-the-art algorithm, showing that the model generates a meaningful community structure. At last, we perform a wide analysis to investigate the impact of a variable community organization on the main structural properties of the synthetic networks.

In Paper C we leverage the nPSO model introduced in Paper B as a benchmark for comparing the performance of community detection and link prediction methods. First, we evaluate state-of-the-art community detection methods using the communities generated by the nPSO model as ground-truth, showing the agreement with previous studies. Second, we test the performance of state-of-the-art link prediction methods on PSO and nPSO networks and verify the importance of community organization for link prediction and the agreement with respect to real networks. Third, we show that the community detection performance can be improved using information gained by the network embedding in the hyperbolic space. At last, we discuss that link prediction in the hyperbolic space still needs further investigation.

Paper A: A. Muscoloni, J. M. Thomas, S. Ciucci, G. Bianconni, C. V. Cannistraci. (2017). Machine learning meets complex networks via coalescent embedding in the hyperbolic space. *Nature Communications*, 8, 1615.

In the previous section, we have reported state-of-the-art methods for embedding a given network in the hyperbolic space. Such methods, provided in input a network topology, infer the radial and angular coordinates of the nodes in the hyperbolic space. The algorithms described, such as the pioneering one of Boguña et al. [18], HyperMap [23] and its extension HyperMap-CN [25], are all based on maximum likelihood approaches, which are generally very time consuming. The time complexity of the embedding methods developed are indeed between O(N^3) and O(N^4), which is still reasonable for networks of small size, but it becomes unfeasible for large systems.

In this paper we show that, adopting topological-based machine learning for nonlinear dimension reduction, the node angular coordinates of the hyperbolic model can be directly approximated in the embedding space according to a persistent node aggregation pattern, which we term ‘angular coalescence’, and we name the related algorithms ‘coalescent embedding’. Manifold machine learning for unsupervised nonlinear dimensionality reduction is an important sub-class of topological machine learning algorithms. They learn nonlinear similarities between points distributed over a hidden manifold in a multidimensional feature space, in order to preserve, map and visualize them in a reduced space [34]. These methods are already used in network biology for instance to predict node similarities in protein interaction networks [35], [36], therefore it was likely for us to envisage their usage for network embedding in the hyperbolic space. We notice that the maximum likelihood approaches previously developed rely on an underlying hyperbolic network model for the inference of both radial and angular coordinates of the nodes. Instead, coalescent embedding algorithms infer the angular coordinates of the nodes in a model-free way, whereas the inference of the radial coordinates is still based on the PSO model theory. Most importantly, the time complexity of the class of algorithms proposed is O(N^2), allowing a fast and accurate embedding also for networks of large size. As a running time example, we embedded large networks of 10000 nodes in less than one minute and 30000 nodes in few minutes.

The mapping accuracy of the hyperbolic embedding methods has been compared on synthetic networks generated with the PSO model, ranging over several combinations of the model parameters. In order to evaluate the mapping accuracy we designed two different measures. The first one is the HD-correlation, which is the Pearson correlation between all the pairwise hyperbolic distances of the network nodes computed from the original coordinates of the PSO model and from the inferred coordinates. This measure provides an overall score of mapping accuracy considering both radial and angular coordinates. The second one is the Concordance score (C-score), which can be interpreted as the proportion of node pairs for which the angular order in the inferred network corresponds to the angular order in the original network. This measure only evaluates the angular ordering of the nodes and therefore their similarity, which are usually the most difficult to infer. Considering both the metrics, we show that the coalescent embedding techniques are outperforming the state-of-the-art methods.

Another important characteristic that can be studied in a network embedded in a geometrical space is its navigability. The network is considered navigable if the greedy routing (GR) performed using the node coordinates in the geometrical space is efficient [16]. In the GR, for each pair of nodes, a packet is sent from the source to the destination and each node knows only the address (coordinates) of its neighbors and the address of the destination, which is written in the packet. In the GR procedure adopted [23], at each hop the packet is forwarded from the current node to its neighbor at the lowest hyperbolic distance from the destination and it is dropped when a loop is detected. In previous works, the GR was generally evaluated computing the percentage of successful paths and the average length of successful paths with respect to the shortest paths. In this paper we designed a new measure for evaluating the efficiency of the GR, which we named GR-score (or GR-efficiency), and it merges both the previous measures in a unique score. The GR-score assumes values between 0, when all the routings are unsuccessful, and 1, when all the packets reach the destination through the shortest path. Comparing the performance of the hyperbolic embedding methods on the PSO networks, we noticed that HyperMap-based algorithms obtained the highest GR-score, followed by the coalescent embedding techniques. The main reason is that the success of the greedy routing is very sensitive to the fact that connected nodes are mapped close in the geometrical space and disconnected nodes far apart. In fact, mapping disconnected nodes close in the geometrical space is likely to cause the routing of packets into wrong paths. While HyperMap maximizes the fact that connected nodes are at low hyperbolic distance and disconnected nodes are at high hyperbolic distance, coalescent embedding put connected nodes close and disconnected far only in the angular coordinates space and not directly in the hyperbolic space, where instead the greedy routing navigation occurs. This point might be improved in future works, designing algorithms still based on fast machine learning techniques but able to optimize hyperbolic distances rather than angular distances only.

Once in possession of fast methods that are able to map complex networks in the hyperbolic space with high precision, several studies can be lead exploiting the geometrical information. Here, as an example of application we show how the hyperbolic distances can be used to feed community detection algorithms. Community structure is a relevant feature of real networks, and consists in the organization of network nodes into groups within which the connections are dense, but between which connections are sparser. The development of algorithms for detection of such communities is a key topic that has broad applications in network science. We modified four approaches for community detection that a recent comparative study has shown to be the best among the state-of-the-art and that accept in input also weighted adjacency matrices. We demonstrate that they can be boosted when applied to the networks weighted according to the hyperbolic distances, which were inferred by our coalescent embedding techniques. The visualization of the inferred coordinates for several real networks also highlights that the communities are well separated over the similarity space of the angular coordinates.

In comparison to the other approaches for hyperbolic embedding developed in previous studies and tailored for the two-dimensional hyperbolic disk, a peculiar property characterizes the class of unsupervised topological-based nonlinear machine learning algorithms adopted here. Being based on matrix decomposition methods for dimensionality reduction, there are not constraints on the number of dimensions that can be used to perform the embedding. Therefore, we investigate the possibility to enlarge the geometrical space from the hyperbolic disk to the hyperbolic sphere, with the addition of a further dimension. The evaluation of the 3D hyperbolic embedding with respect to the 2D embedding for the greedy routing and the community detection applications highlighted that the addition of the third dimension does not lead to a clear and significant improvement in performance. However, further investigations should be provided on networks of larger size and different types of origin, because the 3D space might conceptually offer an advantage with networks of large size. Finally, we want to emphasize that, while the other hyperbolic embedding methods should be re-designed to fit for the three-dimensional space, with the adoption of coalescent embedding approaches the exploration of additional dimensions of embedding is free of charge.

Paper B: A. Muscoloni, C. V. Cannistraci. (2018). A nonuniform popularity-similarity optimization (nPSO) model to efficiently generate realistic complex networks with communities. New Journal of Physics, 20, 052002.

In the previous section we have already introduced the PSO model [20], which describes how random geometric graphs grow in the hyperbolic space optimizing a trade-off between popularity and similarity. In this framework, the popularity of the nodes is represented by the radial coordinate in the hyperbolic disk, whereas the angular coordinates distance is the geometrical counterpart of the similarity between the nodes. Networks generated through the PSO model exhibit strong clustering and a scale-free degree distribution, two among the peculiar properties that usually characterize real-world topologies [11], [37], [38]. However, another important feature commonly observed is the community structure [6], [39], [40], which is lacking in the PSO model, since the nodes are arranged over the angular coordinate space according to a uniform distribution. This issue has been addressed in the study of Zuev et al. [28], introducing the geometric preferential attachment (GPA). The GPA couples the latent hyperbolic network geometry with preferential attachment of nodes to this geometry in order to generate networks with strong clustering, scale-free degree distribution and a non-trivial community structure [28]. Recalling from the previous section, the GPA is characterized by heterogeneous angular attractiveness: the higher the attractiveness of a region the higher the probability that the nodes are placed in that angular section. However, the GPA model does not allow, at least in its current implementation, to directly control in an explicit and efficient way the number and size of the communities, a property that instead might be interesting, for example, while proposing a community detection benchmark. Furthermore, the GPA model does not take into account the possibility to vary the network temperature, an important parameter of the PSO model for tuning the clustering coefficient. For this reasons, in this paper we introduce a variation of the PSO model, which we call nonuniform PSO (nPSO) model, whose key aspects are the possibility of: a) fixing the number and size of communities; b) tuning their mixing property through the network temperature; c) efficiently producing also highly clustered realistic networks.

The procedure to generate a network is the same as for the PSO model, with the only difference that the angular coordinates of the nodes are not sampled uniformly but according to the given nonuniform probability distribution. In particular, without loss of generality, we concentrated on the mixture of distributions where the components are either Gaussian or Gamma distributions, which we consider suitable for describing how to build a nonuniform distributed sample of nodes along the angular coordinates of a hyperbolic disk, with communities that emerge in correspondence of the different components. For instance, given a Gaussian mixture distribution the communities will emerge in correspondence of the different Gaussians, therefore the number of Gaussians can be used to fix the number of desired communities and the mixing proportions of the components can be used to tune their relative size.

One of the main drawbacks to use the original algorithmic implementation (#1) to establish links adopted by the PSO and GPA models also for the nPSO model, is the lack of efficiency in generating networks with communities characterized by high clustering (low temperature), or more in general, with features such that most of the connection probabilities to the targets are low during the network growth. Indeed, the computational time of implementation #1 for generating PSO networks of size N = 1000 is in the order of seconds, whereas for nPSO networks with low temperature it might take up to several hours, depending on the number of communities. Here we propose two other different algorithmic implementations (#2 and #3), showing that both the implementations do not present any issue for generating nPSO networks with low temperature. Analysing the time complexity, both the implementations #1 and #2 have a dependency on the average connection probability to the targets during the generative procedure, which is mainly affected by the temperature and by the extent of the hyperbolic distances. Their complexity formulation is not straightforward to explicitly report since it is function of many parameters of the model. We recommend implementation #3, which resulted the fastest from running time simulations, and whose computational complexity is only dependent on the number of nodes and edges and it is O(EN), requiring 5 minutes to generate large size nPSO networks of N = 10000. In the paper we also provide the mathematical proof that the three implementations generate equivalent network topologies.

As a second result of the paper, we performed a wide investigation on the parameter combinations of the nPSO model for which the emerging communities are detectable by a state-of-the-art algorithm, considering also more complicated settings with asymmetric angular coordinate distributions over the angular space. We highlighted that, for most of the parameter combinations representing realistic scenarios, the community organization can be spotted by the state-of-the-art algorithm Louvain. The main factor that reduces the detectability is the ratio between the number of communities and the network size, in particular community detection in nPSO networks reduces significantly for small size networks that present many communities. These results suggest that realistic community structure is properly reproduced by the model and the nPSO might be employed as a benchmark for testing community detection algorithms, which is the topic of Paper C.

As last analysis, we evaluated and compared several topological measures of the synthetic networks generated using the PSO and nPSO models, and from this wide investigation two important results emerge. First, the parameters of the model allow to reproduce a great variety of the structural properties observed in real-world complex networks, and we provide heatmaps that can be used as a guide for the choice of parameters while generating networks with desired characteristics. Second, the diverse community organization has only a minor impact for most of the main topological measures. This suggests, for example, that if you want to model a given real network and you need to fit the temperature parameter of the model, this can be inferred from the clustering coefficient regardless of the community structure.

Paper C: A. Muscoloni, C. V. Cannistraci. (2018). Leveraging the nonuniform PSO network model as a benchmark for performance evaluation in community detection and link prediction. New Journal of Physics, 20, 063022.

In Paper B we introduced the nPSO model, which grows random geometric graphs in the hyperbolic space, reproducing networks that have realistic features such as high clustering, small-worldness and scale-freeness, with the additional possibility to control the community organization, tuning the number of communities, their size and their mixing. In this paper, we leverage the model as a benchmark for comparing the performance of community detection and link prediction methods.

Providing a benchmark for community detection requires the possibility to manipulate structural properties such as average node degree, clustering, small-worldness and scale-freeness, in order to assess how differently community detection algorithms react to these controlled topological variations. Several generative models have been proposed in the past years as synthetic benchmarks for community detection [40]. One of the first benchmark has been developed by Girvan and Newman [6], where all the nodes have the same degree and all the communities identical size, and has been later extended by Danon et al. [41] in order to generate communities with different size. A generative model able to reproduce structural properties closer to the ones observed in real networks has been proposed by Lancichinetti-Fortunato-Radicchi (LFR) [42], which is characterized by a power-law distribution of degree and community size. A further benchmark for overlapping communities has been introduced by Ball et al. [43], however, having all the nodes the same expected degree, it is less realistic and flexible than the LFR [40]. Interestingly, all these well-known models of networks with communities are not generated according to a latent geometry, which instead is the fundamental theme of this study.

The community detection algorithms Louvain [44], Infomap [45], Walktrap [46] and Label propagation [47] are four state-of-the-art approaches that have been shown to provide high performances on synthetic benchmarks [48]–[50]. Louvain [44] is a model-free and unsupervised heuristic method based on modularity optimization; Infomap [45] finds the community structure by minimizing the expected description length of a random walker trajectory using the Huffman coding process; Walktrap [46] is based on an agglomerative method for hierarchical clustering, where the similarities between the nodes or groups are obtained using random walks; Label propagation [47] is an iterative algorithm in which each node label is updated with the one owned by the majority of the neighbours until reaching a consensus. In this study, we compare these four community detection approaches across synthetic networks generated using diverse nPSO parameter combinations. The results indicate that overall Louvain appears as the strongest approach, with an almost perfect detection over different values of network size, average node degree and temperature. Analysing the advantages and disadvantages of each method over the nPSO parameter combinations, we noticed that the results were in agreement with previous studies [48], in which for example it was shown that for a high mixing of the communities Louvain and Walktrap are more robust, whereas Infomap and Label propagation tend to drop in performance. Therefore, the nPSO model seems to provide a good benchmark to test community detection algorithms on networks generated using a latent geometry model which is based on the hyperbolic space.

The study in Paper A demonstrated that, exploiting the geometrical embedding information in order to weight the adjacency matrix in input to the community detection algorithms, the performance of the respective unweighted variants can be improved [51]. Since the evaluation tests were executed on real network datasets, it remained the doubt that the performance evaluation could have been possibly biased by the restricted metadata available on the community annotation. This is a perfect example for clarifying the utility of the nPSO model. We would like to have a dataset of realistic networks for which we know the ground-truth community organization and that we can use as a benchmark to test whether the community detection algorithms benefit from being applied according to the geometrical embedding information in the hyperbolic space. Therefore, we repeat the same tests of Paper A [51] on real networks, but here we use nPSO synthetic networks as benchmark. More precisely, each network is embedded in the hyperbolic space using the coalescent embedding techniques and the hyperbolic distances between the nodes are used to weight the input network for the Louvain, Infomap, Walktrap and Label propagation algorithms. The results show that almost all the geometrically-weighted methods increase their performance with respect to the unweighted Louvain and Infomap, whereas with the respect to the unweighted Walktrap the increase is relatively small, and with respect to the unweighted Label propagation only some methods increase, but the gain is remarkable. Overall, this investigation on nPSO networks confirms that the performance of community detection algorithms can be improved using network embedding information, in agreement with what has been previously demonstrated on real networks in Paper A [51].

The nPSO can have a large impact not only in the community detection scenario, but also in other real applications such as link prediction. For instance, by generating ground-truth synthetic networks with the nPSO, it is possible to investigate the extent to which the community organization affects, together with other topological properties, the performance of link prediction algorithms. This, in turn, can advocate the comprehension of the intrinsic rules of network wiring that connect topology to geometry and that are encrypted in algorithms for link prediction differentiating their performance. We compared the performance of state-of-the-art approaches [52], [53] (CRA [54]–[56], RA [57], SPM [58], SBM [59]) across real networks, PSO and nPSO networks generated using diverse parameter combinations. Cannistraci-Resource-Allocation (CRA) is a mechanistic model which implements a local-topology-based parameter-free deterministic rule for topological link-prediction motivated by the local-community-paradigm [54]–[56]; the standard Resource-Allocation (RA) is instead motivated by the resource allocation process; Structural Perturbation Method (SPM) is a global and model-free approach that relies on a theory similar to the first-order perturbation in quantum mechanics [58]; Stochastic Block Model (SBM) is a global approach based on general idea of a block model, where the nodes are partitioned into groups and the probability that two nodes are connected depends only on the groups to which they belong [59]. The aim is to understand whether using synthetic networks (generated by PSO or nPSO) is possible to replicate the same and diverse link-predictors' performance obtained on real complex networks. In case this is not possible, we would like to check whether the community organization present in the nPSO plays an important role to replicate the results on real networks and to facilitate the performance of some algorithms in comparison to the case of networks where communities are not present. In the evaluation framework adopted, 10% of the links are randomly removed from the network and used to test the precision of the link predictors in recovering them. The results showed that the ranking of the methods according to their performance on the nPSO model is closer to the one in the real networks with respect to the PSO model, suggesting that the introduction of the communities in the model was important to design a more reliable benchmark for link prediction.

At last, while the previous analysis focused on link prediction approaches exploiting only topological information, we then investigated whether adopting also information gained by the embedding in the hyperbolic space could be beneficial. In particular, we applied the coalescent embedding techniques and the other main hyperbolic embedding methods to map the networks in the hyperbolic space, and we assigned likelihood scores to the nonobserved links based either on hyperbolic distances or hyperbolic shortest paths. The results suggest that the performance is generally lower with respect to the state-of-the-art topological-based link predictors adopted in the previous analysis, and therefore the adoption of geometrical information for link prediction still needs further investigation.

4. Discussion

After having summarized the results of each paper of the dissertation, in this section we will focus on discussing them in perspective of future research. We recall as main results that: in Paper A we have introduced a new class of topological machine learning algorithms for embedding networks in the hyperbolic space, named ‘coalescent embedding’, whose computational complexity is O(N^2); in Paper B we have introduced the nonuniform PSO (nPSO) model, which generates synthetic networks in the hyperbolic space, with high clustering, small-worldness, scale-freeness and a desired community organization, whose computational complexity is O(EN); in Paper C we have shown that the nPSO model can be leveraged as a benchmark for comparing the performance of community detection and link prediction methods.

The discussion section is structured as follows: first, we will review what have been the recent advances on designing methods for embedding networks in the hyperbolic space after the publication of the coalescent embedding; second, we will review the evaluation measures developed until now for assessing the embedding accuracy, we will analyze their differences and highlight points to potentially address in future studies; at last, we will discuss an example of practical application related to a recent study on brain networks.

Since the publication of the coalescent embedding techniques, several other methods for mapping networks in the hyperbolic space have been proposed. Some of them are still based on maximum likelihood algorithms, others on machine learning techniques like the coalescent embedding, others on a mixture of the two approaches, and others faced the problem with completely new ways. Some of these methods will be now discussed.

In a recent preprint of 2019 released on arXiv, Boguña et al. [60] introduced Mercator, a new method for embedding networks in the hyperbolic space that mixes machine learning and maximum likelihood. An attempt of such a mixture approach was already present in a previous study by Alanis-Lobato et al. [61]. Differently from other techniques such as HyperMap [23], which rely on the assumption of the PSO dynamic model [20], this method assumes that the structure of networks is well described by the popularity-similarity S^1/H^2 static geometric network model [15], which can accommodate arbitrary degree distributions and reproduces many pivotal properties of real networks, including self-similarity patterns. The Mercator algorithm is presented with different variants, offering different compromises between running time and mapping accuracy. In brief, in its fast mode, Mercator uses a machine learning technique performing dimensional reduction to produce a fast and accurate map, similarly to one variant of the coalescent embedding class of algorithms, but replacing some of the algorithmic steps with adjustments based on the popularity-similarity S^1/H^2 static model. In the refined Mercator mode, instead, the fast mode embedding result is taken as an initial condition in a maximum likelihood estimation, which improves the quality of the final embedding, although increasing the running time, with a complexity of (E^2) [60].

In another recent preprint of 2019 released on arXiv, Keller-Ressel and Nargang [62] introduced hydra (hyperbolic distance recovery and approximation), a novel method for embedding network or distance-based data into hyperbolic space, which has strong mathematical foundations and does not depend on specific assumptions on network growth or structure. At the same time, the method is computationally efficient and based on reduced matrix Eigendecomposition [62]. The authors show mathematically that, when presented with mutual distances of data points located on a low-dimensional hyperbolic submanifold of the feature space, hydra will recover these points exactly [62]. For general data, the method satisfies a certain optimality property, similar to the strain-minimizing property of multidimensional scaling. If minimization of stress is the ultimate objective and strain is used only as a proxy, the result of hydra can be used as an initial condition for a direct minimization of the stress functional, for which efficient quasi-Newton routines can be used. The authors name such extension as hydra+ [62].

In 2018, Bläsius et al. [63] designed and implemented a new algorithm for computing hyperbolic embeddings of large-scale networks based on maximum likelihood estimation. Compared to previous approaches that needed $\Omega(N^2)$ runtime, their algorithm runs in quasilinear runtime. As first modification, they used an analytical approach to compute the expected angular distances between pairs of high-degree nodes based on their number of common neighbors. In contrast to HyperMap-CN [25], this approach does not rely on expensive numerical computations, making it fast in practice. The resulting angle distance matrix is then fed to a spring embedder that finds good positions for high-degree nodes in linear time. For small degree nodes, the runtime is substantially improved by using the geometric data structure of Bringmann et al. [64] that allows traversing nodes of close proximity in expected amortized constant time. This enables to embed significantly larger graphs than before, requiring for instance less than one hour to compute a hyperbolic embedding of the Amazon product recommendation network with over 300000 nodes [63].

In a more recent preprint in collaboration between Cannistraci and me, released on arXiv [65], we proposed a completely new approach, neither based on machine learning nor on maximum likelihood estimation. In this work we present a mechanistic model named minimum curvilinear automaton (MCA), which relies on a mechanism of node growth that we named similarity attachment, and that allows an efficient inference of the angular coordinates for the embedding of networks in the hyperbolic disk [65]. The revolutionary idea is that a local rule of node attachment implements a network automaton that grows according to a strategy of network navigation termed minimum curvilinearity. The idea of minimum curvilinearity (MC) is that the hidden geometry of complex networks that are in particular sufficiently sparse, clustered, small-world and heterogeneous can be efficiently navigated using the minimum spanning tree (MST), which is a greedy navigator. The local topological information drives the global geometrical navigation and the MST can be interpreted as a growing path that greedily maximizes local similarity between the nodes attached at each step by globally minimizing their overall distances in the network [65]. Here we show that, according to the mechanism of node similarity attachment, the visited node sequence of a growing MST can efficiently approximate the node angular coordinates in the hyperbolic disk, that actually represent an ordering of their similarities. This is a consequence of the fact that the MST, during its greedy growing process (for instance adopting the Prim's algorithm [66]), at each step sequentially attaches the node most similar (less distant) to its own cohort. The network should be firstly pre-weighted in order to guide the MST growth according to a "good guess" of the edge weights that suggest the geometrical distances. MCA displays an embedding accuracy (evaluated using the HD-correlation) that in general seems superior to HyperMap-CN and inferior to coalescent embedding. However, the network mappings of coalescent embedding are less navigable (lower GR-score) than the ones of MCA, which appears therefore as the most robust over the different evaluation frameworks [65]. Remarkably, MCA is also the best method for link prediction on many diverse real networks of different size and topological characteristics. Depending on the type of minimum edge-weight data structure used to grow the MST, the MCA time complexity can also approach a linear dependence from the number of edges [65].

This review about the recent advances on the development of hyperbolic embedding methods highlighted that the field is still very active in trying to find the algorithm providing the best compromise between computational time complexity and mapping accuracy. However, it should be noted that what mapping accuracy means is still not well defined. In Paper A we have introduced the HD-correlation [51], which is a measure to compare all the pairwise hyperbolic distances of the original coordinates generated by the model with respect to the inferred ones. This measure evaluates the mapping accuracy adopting a synthetic model as ground-truth and considering both radial and angular coordinates, since hyperbolic distances include both of them. In Paper A we have also designed the C-score [51], which instead evaluates the ordering of the nodes over the similarity space. This measure evaluates the mapping accuracy adopting a synthetic model as ground-truth, but only considering the angular coordinates, and in particular their ordering. Another evaluation measure is the GR-efficiency [51], [67], which assesses the network navigability according to the greedy routing protocol. This measure evaluates the mapping accuracy without requiring a synthetic model as ground-truth, taking into consideration hyperbolic distances (therefore both radial and angular coordinates), but in relation to a very specific task, greedy navigability, which does not necessarily characterize all the real networked systems. Other quantitative evaluations can be performed in an application-oriented framework, such as link prediction or community detection. In this scenario the evaluation of the mapping accuracy would not require a synthetic model as ground-truth, however, like for the greedy navigability, it would be related to a specific task. In summary, there are many different ways of evaluating the mapping accuracy, and not always they are strongly related between each other. Therefore it is possible that some hyperbolic embedding methods obtain higher performance while tested on a certain framework, for example link prediction, and others instead result superior under another framework, for example greedy navigability. Given this, the field might benefit from a wide comparative study analyzing all the hyperbolic embedding methods developed until now, testing them under several evaluation frameworks and highlighting their computational complexity, trying to point out for each framework the method offering the best performance in a reasonable running time.

To conclude, we will discuss an example of practical application related to a recent study on brain networks, released as a preprint on arXiv in 2017 by Cacciola et al. [68], in which I share the co-first authorship. In structural connectomes, topological patterns of connectivity are often related with the physical distances between elements of the brain networks. On one hand, brain regions that are spatially close have a relatively high probability of being interconnected, on the other hand, longer white matter projections are more expensive in terms of their material and energy costs, thus making connections between spatially far brain structures less likely [69]. As a first result of the work, Cacciola et al. [68] demonstrate that, by applying coalescent embedding techniques to map structural brain connectomes of healthy subjects, they can unsupervisedly reconstruct the anatomical arrangement of the brain regions. In other words, the nodes of the brain networks, representing structural brain regions, are arranged in the similarity space of the hyperbolic embedding with an ordering that resembles the left and right hemispheres of the brain, and, more in details, the lobes organization (frontal, parietal, occipital, temporal). This result demonstrates a strong relationship between topology and geometry of brain structural connectomes. As second analysis of the study, Cacciola et al. [68] investigated whether adopting network markers based on the hidden metric space behind the brain networks, it would be possible to detect connectomic variations between healthy controls and de novo drug naïve Parkinson’s Disease patients. The analysis have been repeated also comparing two groups of healthy controls within different age ranges. The results show interesting trends, according to which several latent geometry network markers, computed for example as the average of the pairwise distances in the hyperbolic embedding space, are able to significantly detect the separation between the two groups [68]. Such results, which require further investigation testing additional datasets and pathological conditions, pave the way for practical applications of hyperbolic embedding methods in network medicine.

Part II: Papers of the dissertation

Paper A: Machine learning meets complex networks via coalescent embedding in the hyperbolic space

Alessandro Muscoloni[1,†], Josephine Maria Thomas[1,†], Sara Ciucci[1,3], Ginestra Bianconi[4] and Carlo Vittorio Cannistraci[1,2,*]

[1] Biomedical Cybernetics Group, Biotechnology Center (BIOTEC), Center for Molecular and Cellular Bioengineering (CMCB), Center for Systems Biology Dresden (CSBD), Department of Physics, Technische Universität Dresden, Tatzberg 47/49, 01307 Dresden, Germany; [2] Brain bio-inspired computing (BBC) lab, IRCCS Centro Neurolesi “Bonino Pulejo”, Messina, Italy; [3] Lipotype GmbH, Tatzberg 47, 01307 Dresden, Germany; [4] School of Mathematical Sciences, Queen Mary University of London, London E1 4NS, United Kingdom; * Correspondence should be addressed to: kalokagathos.agon@gmail.com; † The first two authors should be regarded as joint First Authors

Published: *Nature Communications*, 8, 1615 (2017).

Abstract

Physicists recently observed that realistic-complex-networks emerge as discrete samples from a continuous hyperbolic-geometry enclosed in a circle: the radius represents the node centrality and the angular displacement between two nodes resembles their topological proximity. The hyperbolic-circle aims to become a universal space of representation and analysis of many real-networks. Yet, inferring the angular-coordinates to map a real network back to its latent geometry remains a challenging inverse-problem.

Here we show that intelligent-machines for unsupervised recognition and visualization of similarities in big-data can also infer the network angular-coordinates of the hyperbolic-model according to a geometrical organization that we term ‘angular-coalescence’. Based on this phenomenon, we propose a class of algorithms that offers fast and accurate ‘coalescent-embedding’ in the hyperbolic-circle even for large-networks. This computational solution to an inverse-problem in physics of complex-systems favours the application of network latent-geometry techniques in disciplines dealing with big-network-data analysis including biology, medicine and social science.

A.1 Introduction

Significant progress has been achieved in the last twenty years in unveiling the universal properties of complex networks. Nevertheless, the characterization of the large variety of real network structures, which are originating from the ‘Big Data explosion’, remains an important challenge of network science. Network geometry aims at making a paradigmatic shift in our understanding of complex network structures by revealing their hidden metric [15], [16], [20], [35], [36], [54], [55], [70]–[72]. This field has a large number of applications ranging from brain networks [54] to routing packets in the Internet [18], [73]. In this context there is increasing evidence that the hidden metric of many complex networks is hyperbolic [74]. Examples of recent research topics are the development of tools to generate hyperbolic networks [75], [76], the measurement of the hyperbolicity of complex networks [77], [78], the analysis of its impact on traffic congestion [79], [80] and on link prediction [81], the characterization of network properties in terms of the parameters of hyperbolic network models [82] and the study of time-varying control systems with hyperbolic network structure [83]. However, the science that studies and designs algorithms to reveal and to test the latent geometry [29] of real complex networks, is in its dawning.

The Popularity Similarity Optimization (PSO) model suggests that real networks have a congruous geometrical representation in a hyperbolic space, where each network node is mapped according to the angular and the radial coordinates of a polar system [20]. On one hand, node similarities are related with the angular distances in the hyperbolic space: the higher the similarity between two nodes, the closer their angular coordinates. On the other hand, the node degree is related with the intrinsic popularity of the node: the higher the node degree, the higher its popularity in the network and the lower its radial coordinate in the hyperbolic space. Recently, further variants of the PSO model have been proposed in order to produce hyperbolic synthetic networks with soft communities [28] or with a desired community structure [84].

Manifold machine learning for unsupervised nonlinear dimensionality reduction is an important sub-class of topological machine learning algorithms. They learn nonlinear similarities/proximities (that can be also interpreted as dissimilarities/distances) between points (samples) distributed over a hidden manifold in a multidimensional feature space, in order to preserve, embed (map) and visualize them in a two-dimensional reduced space [34]. They are inspired by a three-step procedure. First, they approximate the shape of the hidden manifold reconstructing a nearest-neighbourhood graph between the points in the high-dimensional space. Second, they use the reconstructed network to estimate pairwise topological similarities (or distances) between the points that lie on the manifold, and store these nonlinear estimations in a kernel (or distance matrix). In a third and last step, they apply a matrix decomposition to the kernel to perform dimensionality reduction, usually in a space of two dimensions. If the network is already given in the form of an unweighted adjacency matrix, the same algorithm works neglecting the first step and thus, in practice, performs a network embedding that preserves the node similarities. These methods are already used in network biology for instance to predict node similarities in protein interaction networks [35], [36], therefore it was likely for us to envisage their usage for network embedding in the hyperbolic space.

Here we show that, adopting topological-based machine learning for nonlinear dimension reduction, the node angular coordinates of the hyperbolic model can be directly approximated in the two or three dimensional embedding space according to a persistent node-aggregation pattern, which we term 'angular coalescence'. Based on this phenomenon, we propose a class of algorithms that offers fast (time complexity approximately $O(N^2)$, with N indicating the network node size) and accurate 'coalescent embedding' in the two or three dimensional hyperbolic space even for large unweighted and weighted networks. This discovery paves the way for the application of network latent-geometry techniques in many disciplines dealing with big-network-data analysis including biology, medicine and social science.

A.2 Results

Coalescent embedding

In this study we selected a representative group of nonlinear topological-based unsupervised dimensionality reduction approaches among the ones with the highest performance. Three manifold-based: Isomap (ISO) [85], noncentered Isomap (ncISO) [36], Laplacian eigenmaps (LE) [86]. Two minimum-curvilinearity-based: minimum curvilinear embedding (MCE) [34], [36] and noncentered minimum curvilinear embedding (ncMCE) [36]. An important note for practical applications is that these approaches are unsupervised (node or edge labels are not required) and parameter-free (external tuning setting of algorithms' parameters is not required). In Fig. 1B-D we show the embedding provided by the Isomap algorithm (ISO), the progenitor of the manifold dimension reduction techniques, starting from the unweighted adjacency matrix of a PSO network. The nodes are organized according to a circular pattern (Fig. 1B), which follows the angular coordinates of the original PSO model. For an algorithm named noncentered minimum curvilinear embedding (ncMCE) [36] (Fig. 1E-F), the circular pattern is linearized (Fig. 1E) and the nodes are ordered along the second dimension of embedding according to their similarities. If we accommodate the node points on the circumference following the same ordering as the second dimension of embedding (Fig. 1E), we can again recover an unbroken circular pattern (Fig. 1F) that resembles the angular coordinates of the original PSO model. The ability of ncMCE and minimum-curvilinearity-based algorithms to learn, unfold and linearize along just one dimension an intrinsic nonlinear (circular) pattern is discussed in details in the Methods section. However, here we clarify that minimum-curvilinearity-based algorithms compress the information in one unique dimension because they learn nonlinear similarities by means of the minimum spanning tree (MST), providing a hierarchical-based mapping that is fundamentally different from the manifold-based of ISO.

The rationale of our approach is all contained in these simple insights. We embedded hyperbolic networks adopting different combinations of network similarities and matrix decompositions and we reached the same consistent finding: the arising in the two-dimensional embedding space of a common node aggregation pattern, which we named 'angular coalescence', and that was circularly or linearly ordered according to the angular coordinates of the hyperbolic model. This represents the first important result of our study. The term 'angular coalescence' is proposed to indicate that the individual nodes aggregate together (from the Latin verb coalēscō: to join, merge, amalgamate single elements into a single mass or pattern) forming a pattern that is progressively ordered along the angular dimension. Consequently, we decided to coin the expression 'coalescent embedding' to indicate the class of algorithms that exhibit angular coalescence in the two-dimensional network embedding. In our case we detected the angular coalescence phenomenon as embedding result of topological-based machine learning for nonlinear unsupervised dimension reduction. Indeed, the evidence that even MCE and ncMCE, which are not manifold-based but hierarchical-based, are able to exhibit coalescent embedding may theoretically suggest that this is an 'epiphenomenon' that in general characterizes topological-based machine learning for nonlinear dimension reduction when applied to this task.

Given the first results, we propose to adopt these machine learning techniques to perform two-dimensional 'structural network imaging', which could be figuratively envisaged as a sort of in-silico imaging technique (such as X-ray or MRI are for condensed matter) for 2D-reconstruction and visualization of the hidden manifold shape from which the structural organization of a complex network emerges.

In the Methods section we propose a general algorithm - based on the angular coalescence principle - for network embedding in the hyperbolic space. In Fig. 2 a flow chart is reported, where both the algorithmic steps and the intermediate input/output are highlighted. In order to build a general algorithm we started by noticing that the problem to compute the embedding on an unweighted adjacency matrix would be simplified by having a ‘good guess’ of the edge weights that suggest the connectivity geometry. Thus, there was a clear margin to improve the coalescent embedding performance by pre-weighting the network links using a convenient strategy to approximate distances between the connected nodes. We devised two different pre-weighting strategies. The first approach - which we called the repulsion-attraction rule (RA) – assigns an edge weight adopting only the local information related to its adjacent nodes (neighbourhood topological information). The idea is that adjacent nodes with a high external degree (where the external degree is computed considering the number of neighbours not in common) should be geometrically far because they represent hubs without neighbours in common, which - according to the theory of navigability of complex networks presented by Boguñá et al. [16] - tend to dominate geometrically distant regions: this is the repulsive part of the rule. On the contrary, adjacent nodes that share a high number of common neighbours should be geometrically close because most likely they share many similarities: this is the attractive part of the rule. Thus, the RA (see Equation 1 and 2 for two alternative mathematical formulations) is a simple and efficient approach that quantifies the trade-off between hub repulsion and common-neighbours-based attraction. Supplementary Fig. 1 gives a visual example about how the RA pre-weighting rule is improving the angular coalescence effect with respect to Fig. 1, where the same methods are adopted without pre-weighting. Since it might be argued that the repulsion between high external degree nodes implied by the RA rule is in contrast with the existence of rich-clubs, in Supplementary Discussion we comment the rich-clubness of the PSO networks (Supplementary Fig. 25) and why this does not affect the RA pre-weighting efficiency. Although inspired by the same rationale, the second strategy makes, instead, a global-information-based pre-weighting of the links, using the edge-betweenness-centrality (EBC) to approximate distances between nodes and regions of the network. EBC is indeed a global topological network measure, which expresses for each edge of the network a level of centrality, and the assumption is that central edges are bridges that tend to connect geometrically distant regions of the network, while peripheral edges tend to connect nodes in the same neighbourhood. We let notice that if a weighted network is given, where the weights suggest distances between connected nodes, these can be directly adopted rather than approximated by the pre-weighting techniques.

Furthermore, we were not convinced that preserving the angular distances between nodes adjacent in the angular coordinates was the best strategy. Most likely their reciprocal angular distances were affected by short-range angular noise. Thus, we devised a strategy to re-organize the nodes on the circumference that we called equidistant adjustment (EA): the nodes are equidistantly re-organized along the angular coordinates of the circumference according to their original order learned by the coalescent embedding. Fig. 2 displays a didactic example of the difference between the circular and equidistant adjustment.

The several variants of coalescent embedding algorithms, characterized by the different pre-weightings and angular adjustments, have been tested in various evaluation frameworks using both synthetic and real networks, and their performance has been compared to state of the art methods for hyperbolic embedding. The next sections will report the results obtained together with a discussion of the main achievements.

Evaluations of mapping accuracy in PSO synthetic networks

In order to test the performance of the hyperbolic embedding methods, synthetic networks have been generated with the PSO model, ranging over several combinations of the parameters. Fig. 3 reports the results of the best dimension reduction methods for the first evaluation framework. Here the performance was evaluated as Pearson correlation between all the pairwise hyperbolic distances of the network nodes (we called such correlation: HD-correlation) in the original PSO model and in the reconstructed hyperbolic space. The plots report the average correlation over the 100 synthetic networks that have been generated for each different PSO model parameter combination. It is evident that the coalescent embedding techniques pre-weighted with RA and adjusted according to EA are outperforming HyperMap [23], HyperMap-CN [25] and LPCS [87] that are the state of the art, and this is the second key discovery of our study. RA performed similarly to EBC, and in general both the pre-weighting strategies are effective (Supplementary Fig. 2-6). However, RA is computationally more efficient because it is a local approach (see Methods for details about the complexity). Obviously, all the methods reduce their performance for increasing temperature (reduced clustering), because the networks assume a more ‘random’ structure.

Another alluring result, pointing out a very subtle problem, is that without EA all the techniques significantly reduce the performance, as it is shown in Supplementary Fig. 7. Looking at Fig. 3 and the Supplementary Fig. 2-6, EA makes a difference especially for low temperatures (high clustering), while for high temperatures its improvement is vanishing. This is particularly evident for LE that in Supplementary Fig. 7 (where EA is not applied) at low temperatures has a significantly worse performance compared to Fig. 3 where EA is applied. Imposing an equidistant adjustment might be counterintuitive, but our simulations suggest that this sub-optimal strategy is better than passively undergo the short-range angular embedding uncertainty. On the other hand, once the temperature is increased, the overall angular embedding uncertainty also increases and the techniques are less efficient to recover the node order. In practice, for high temperatures the overall noise overcomes the short-range noise and the EA reduces its effectiveness.

In Fig. 4B, we repeated the same evaluation of Fig. 3, but we adopted a different measure called Concordance score (C-score). The C-score can be interpreted as the proportion of node pairs for which the angular relationship in the inferred network corresponds to the angular relationship in the original network (see Methods). Basically, this score provides a quantitative evaluation of the scatter plots in Fig. 4A where the alignment between inferred and original angular coordinates is visually compared. A C-score of 0 indicates total misalignment, while 1 indicates perfect alignment. The results in Fig. 4 confirm that our methods outperform, especially for low temperatures, the state of the art techniques also in recovering a good angular alignment of the nodes. Supplementary Fig. 8-17 show the scatter plots for all the other methods and temperatures. The scatter plots visually highlight that the correlation between real and inferred angular coordinates decreases for increasing temperatures. However, it is evident that the proposed techniques are able to provide quite accurate alignments even for middle temperatures. Noticeably, the coalescent-embedding-based algorithms combine important performance improvement with a spectacular speed up in respect to HyperMap (Fig. 5 and Supplementary Fig. 18). They can even embed large networks of 10000 nodes in less than one minute and 30000 nodes in few minutes (Fig. 5), whereas HyperMap requires more than three hours for small networks of just 1000 nodes. It is important to underline that, in addition to the remarkable scaling of the computational time (see Methods for details about the complexity), the high correlation values are also preserved for larger networks.

Greedy routing performance in synthetic and real networks

Another important characteristic that can be studied in a network embedded in a geometrical space is its navigability. The network is considered navigable if the greedy routing (GR) performed using the node coordinates in the geometrical space is efficient [16]. In the GR, for each pair of nodes, a packet is sent from the source to the destination and each node knows only the address (coordinates) of its neighbours and the address of the destination, which is written in the packet. In the GR procedure adopted [23], at each hop the packet is forwarded from the current node to its neighbour at the lowest hyperbolic distance from the destination and it is dropped when a loop is detected. The efficiency is evaluated according to the GR-score (see Methods for details), which assumes values between 0, when all the routings are unsuccessful, and 1, when all the packets reach the destination through the shortest path. Supplementary Fig. 19 compares the performance of the hyperbolic embedding methods as mean GR-score over all the PSO networks of Fig. 3. While mean GR-score on 8 real networks (whose statistics are reported in Table 1) are shown in Fig. 6. The first fact to notice is that the PSO network as synthesized with its original coordinates is the most navigable network. Secondly, HyperMap-based algorithms obtained the highest GR-score among the hyperbolic embedding algorithms, followed by the coalescent embedding technique RA-ncMCE, which turns out to be the best both considering EA and non-EA versions. However, the mean GR-score of RA-ncMCE in Fig. 6 is not statistically different from the one of the HyperMap-based algorithms (permutation test p-value > 0.2 in all the pairwise comparisons), therefore their performance is comparable on real networks of Fig. 6. This is an impressive result and we will now explain the reason. The success of the greedy routing is very sensitive to the fact that connected nodes are mapped close in the geometrical space and disconnected nodes far apart. In fact, mapping disconnected nodes close in the geometrical space is likely to cause the routing of packets into wrong paths. In the original PSO network, nodes are connected with probability inversely proportional to their hyperbolic distance [20], therefore connected nodes tend to be close and disconnected nodes faraway by construction, which explains the high navigability of the networks generated with the PSO model. The reason why HyperMap methods offer the best GR performance is that - during maximum likelihood estimation procedure - they iteratively adjust both the angular and radial coordinates of the nodes using an objective function that is maximized if connected nodes are at low hyperbolic distance and disconnected nodes are at high hyperbolic distance [23]. The reason why coalescent embedding techniques offer a GR performance that is inferior to HyperMap methods is that they put connected nodes close and disconnected far only in the angular coordinates and not directly in the hyperbolic space, where instead the greedy routing navigation occurs. In brief, coalescent embedding optimizes angular distances in order to put connected points close and disconnected far, while HyperMap optimizes the hyperbolic distances. Therefore, the results obtained by RA-ncMCE in greedy routing are impressive considering that for this method only angular coordinates contribute to the organization of the points in the hyperbolic space, and that despite this significant limitation RA-ncMCE performances on real networks are comparable to the ones of HyperMap methods. This finding is promising since further algorithms might also be designed to embed directly in the hyperbolic space instead of inferring exclusively angular coordinates, as for the moment coalescent embedding is able to do. A digression on the reason why RA-ncMCE is the best performing among the coalescent embedding methods is provided in Supplementary

Discussion, together with an analysis on the impact of the equidistant adjustment for greedy routing, reported in Supplementary Fig. 20.

Community detection on real networks

Once in possession of fast methods that are able to map complex networks in the hyperbolic space with high precision and to disclose the hidden geometry behind their topologies, several studies can be lead exploiting the geometrical information. The analyses can cover disparate fields like social, communication, transportation, biological and brain networks. As an example of application we show how the hyperbolic distances can be used to feed community detection algorithms. Community structure is one of the most relevant features of real networks, and consists in the organization of network nodes into groups within which the connections are dense, but between which connections are sparser. The development of algorithms for detection of such communities is a key topic that has broad applications in network science, for example in identifying people with similar interests in social networks, functional molecular modules in biological networks or papers with related topics in citation networks [6]. We modified four approaches for community detection that a recent comparative study [48] has shown to be the best among the state of the art and that accept in input also weighted adjacency matrices: Louvain [44], Infomap [45], Label propagation [47] and Walktrap [46]. We demonstrate that they can be boosted when applied to the networks weighted according to the hyperbolic distances, which were inferred by some of our coalescent embedding techniques (see Methods for details). In general, our results show that, regardless of the approach used for community detection, ncISO-based and MCE-based coalescent embedding techniques are significantly better than LE-based in this task on real networks (Table 1-2 and Supplementary Table 1-4). The improvement obtained for Infomap is moderate but very reliable: indeed EBC-ncISO-EA allows always (on every network) the improvement or the same performance in respect to standard Infomap (unweighted). The boost obtained for Louvain is remarkable (but less stable), indeed EBC-ncISO-EA, which is also here the best method, offers an overall improvement of +13.2%. In particular, an astonishing performance is obtained for a social network, the Karate Club [88] (Table 1, first column), where the Louvain algorithm based on the EBC-ncISO-EA embedding reaches the perfect community detection - a result that is evident also in the hyperbolic space visualization (Fig. 7A) - whereas the unweighted Louvain, Infomap, Label propagation and Walktrap algorithms on the same network attain a mediocre performance. The Karate network represents the friendship between the members of a university karate club in US: communities are formed by a split of the club into two parts, each following one trainer.

This is a valuable pedagogic result, indeed to the best of our knowledge it is the first time that the communities present in the Karate network are perfectly untangled by means of the Louvain algorithm (which is ineffective without the ‘geometrical’ boost of the coalescent embedding), and few algorithms can achieve these results in general [89]. On the other hand, significant Louvain improvements are obtained for the majority by the MCE-based approaches (Table 1): in Fig. 7B-D we offer some real network examples of the embedding efficacy of these techniques for disclosing and visualizing the communities directly in the hyperbolic space. We gained perfect community detection also for another type of social network of larger size (composed of four hidden communities), the Opsahl_11 [90] (Fig. 7C). This is a type of intra-organisational network where a link indicates that the connected employees have both awareness of each other’s knowledge and skills on the job. The four hidden communities are related with the location of the employers (Paris, Frankfurt, Warsaw and Geneva) and they were perfectly detected starting from the social-based network topology (Fig. 7C). For the Label propagation and Walktrap algorithms the presence of a performance improvement given by some coalescent embedding techniques is confirmed (Supplementary Table 3-4) and most of them are again ncISO-based and MCE-based approaches. Further discussions on the impact of the equidistant adjustment and on the results for large-size real networks are provided in Supplementary Discussion and Supplementary Table 6-8.

Beyond the two-dimensional space

In comparison to the other approaches for hyperbolic embedding developed in previous studies and tailored for the two-dimensional hyperbolic disk, a peculiar property characterizes the class of unsupervised topological-based nonlinear machine learning algorithms adopted here. Being based on matrix decomposition methods for dimensionality reduction, there are not constraints on the number of dimensions that can be used to perform the embedding. This led us to investigate the possibility to enlarge the geometrical space from the hyperbolic disk to the hyperbolic sphere, with the addition of a further dimension.

Therefore, we have adopted the manifold-based unsupervised machine learning algorithms (LE, ISO, ncISO) in order to extend the coalescent embedding to the three-dimensional hyperbolic space. After the pre-weighting step, the nonlinear dimension reduction is performed using an additional dimension with respect to the two-dimensional case. Considering a spherical coordinate system, the polar and azimuthal angles of the nodes in the hyperbolic sphere are preserved from the dimensionally reduced coordinates, whereas the radial coordinate is assigned as for the hyperbolic disk. The minimum-curvilinearity-based algorithms as well as the equidistant adjustment are not suitable for this extension, detailed explanations are provided in Supplementary Discussion. The analysis of the 3D hyperbolic embedding of the PSO networks highlighted the presence of common patterns, for which Fig. 8 and Supplementary Fig. 21 show an explanatory example. At low temperature ($T = 0$) the nodes appear distributed over a well-defined closed 3D curve. Intuitively, it seems that the 2D hyperbolic disk already offers a perfect discrimination of the similarities and with the addition of the third dimension there is not much gain of information. With the increase of the temperature the nodes look more and more spread around the closed 3D curve that was well-defined at low temperature. Even if one angular dimension of the sphere still recovers most of the similarities present in the original network, it is unknown if the higher spread along the second angular dimension consists in a more refined discrimination between similar nodes or in noise. Since the original coordinates are two-dimensional, this cannot be easily assessed.

In order to analyse the quality of the mapping from a quantitative point of view, the improvement given by the 3D hyperbolic embedding with respect to the 2D embedding is evaluated for the greedy routing and the community detection applications, the results are shown in Supplementary Table 9-12 and commented in details in Supplementary Discussion.

Overall, the tests both on real and artificial networks represent a quantitative evidence that the addition of the third dimension of embedding in the hyperbolic space does not lead to a clear and significant improvement in performance. Although for the PSO model this is indeed expected (because the synthetic networks are generated by a 2D geometrical model), we obtain the same result also on real networks, for which the hidden geometry is not necessarily 2D. Therefore, we might conclude that in practical applications, at least on the tested networks, the two-dimensional space appears to be enough for explaining the hidden geometry behind the complex network topologies. However, further investigations should be provided on networks of larger size and different types of origin, because the 3D space might conceptually offer an advantage with networks of large size. An additional interesting test can be to generate synthetic networks using a 3D PSO model, and then to compare the embedding accuracy using mapping techniques in 2D and 3D. Finally, we want to emphasize that, while the other hyperbolic embedding methods should be re-designed to fit for the three-dimensional space, with the adoption of coalescent embedding approaches the exploration of additional dimensions of embedding is free of charge.

A.3 Discussion

The investigation of the hidden geometry behind complex network topologies is an active research topic in recent years and the PSO model highlighted that the hyperbolic space can offer an adequate representation of the latent geometry of many real networks in a low dimensional space. However, in absence of a method able to map the network with high precision and in a reasonable computational time, any further analysis in the geometrical space would be compromised. Here we propose coalescent embedding: a class of unsupervised topological-based nonlinear machine learning algorithms for dimension reduction that offer a fast and accurate embedding in the hyperbolic space even for large graphs, this is the main product of the work. The embedding methods can be at the basis of any kind of investigation about network geometry and, as examples of applications, we presented community detection and greedy routing. However, the impact of coalescent embedding can be of importance for many disciplines including biology, medicine, computer science and physics.

Below, we will summarize the main findings of this study. The first is that coalescent embedding significantly outperforms existing state of the art methods for accuracy of mapping in the hyperbolic space and, at the same time, reduces the computational complexity from $O(N^3)$ - $O(N^4)$ of current techniques to $O(N^2)$. In addition, the results obtained on synthetic networks are indicative but should be considered with caution. In fact, LE-based coalescent embedding that performs better on synthetic networks is clearly outperformed in real network applications by MCE-based coalescent embedding. This implies that real networks might have a geometry that is even more tree-like and hyperbolic (for this reason MCE-based techniques can perform better on real networks) than the one hypothesized by the PSO model with uniform probability distribution of angular coordinates. In addition, although the topology of many real networks is certainly conditioned by the hyperbolic geometry this is however one of the factors that shape their structure. Interestingly, good results are achieved also for networks with out of range γ values. Since it has been demonstrated that a scale-free degree distribution is a necessary condition for hyperbolic geometry [17], this result demonstrates that the coalescent embedding methods can reach good performances also for networks whose latent geometry might be weakly-hyperbolic or not hyperbolic.

The second important result is that the greedy routing performance on real networks embedded in the hyperbolic space using RA-ncMCE (which is a special type of coalescent embedding based on Minimum Curvilinearity) is only slightly inferior and in general comparable (no significant statistical difference is detected) to the one of networks mapped using the state of art methods. This is a remarkable result because the state of the art methods directly optimize the hyperbolic distances in order to map connected nodes close and disconnected nodes far, which is a key factor for effective greedy routing. RA-ncMCE, instead, offers a comparable performance by ordering only the angular coordinates, which is a promising starting point to develop more effective strategies. In addition, RA-ncMCE can provide embedding of networks with 30000 nodes in few minutes (Supplementary Fig. 22), hence can be of crucial aid for the investigation of greedy routing in real large networks. However, this last claim needs further analysis, because using the state of the art methods on networks of 30000 nodes was not computationally achievable in this study, therefore we cannot ensure that also on large networks RA-ncMCE provides a performance comparable to state of the art methods such as HyperMap.

The third important finding of this study is that coalescent embedding can improve the performance of several state of the art community detection algorithms when they are applied to real networks of small and medium size. But this improvement tends to be of less entity for large networks.

The previous state of the art methods such as HyperMap were tailored for mapping networks in the hyperbolic space. The fourth key achievement of this study is that coalescent embedding techniques, although here adopted and tested for the inference of hyperbolic angular coordinates, are topological-based machine learning algorithms for nonlinear dimensionality reduction and therefore theoretically able to unfold any network latent-geometry, not necessarily hyperbolic. The main point consists in the ability to design or learn a kernel that approximates the geometry of the hidden manifold behind the network topology. To this aim, we proposed two network pre-weighting node-similarity methods (RA that is local-topology-based and EBC that is global-topology-based) aimed to approximate link geometrical information starting from the mere network topology. However, at the moment we are making the first steps towards understanding these mechanisms of self-organization in complex networks, only further scientific efforts will help in the years to come to take advantage of the promising solutions proposed in this article to address many other questions in network geometry. For instance, the hidden geometry of networks with strong clustering and a non-scale-free degree distribution has been demonstrated to be Euclidean [29]. Thus, for networks with these characteristics, the Euclidean embedding obtained by the dimension reduction could be in theory directly adopted rather than exploiting it for the inference of the hyperbolic angular coordinates. However, this is just a speculation that needs to be proved in future studies. The fifth and last significant discovery of this study consists in an innovation introduced with coalescent embedding techniques that, although adopted for the mapping on a two-dimensional hyperbolic space, offers the possibility to explore spaces of higher dimensionality. Our simulations showed that the coalescent embedding on a three-dimensional hyperbolic sphere does not lead to a significant improvement on the tested datasets for tasks such as greedy routing and community detection. However, this result does not prevent further investigations that can lead to different results when the method is employed for embedding in arbitrary dimensions on new data and with different aims. Although in this study only unweighted networks have been used, we let notice that the coalescent embedding methods can work also on weighted networks, where the weights suggest geometrical distances between the connected nodes, and this is a further advantage with respect to state of the art methods such as HyperMap. We want to stress that the exploitation of these machine learning techniques in complex network latent-geometry is, as a matter of fact, initiated in this article, hence we suggest to take all the results here presented with a 'grain of salt'. Nevertheless, gathered all together our findings suggest that the idea to connect unsupervised topological-based nonlinear machine learning theory with network latent-geometry is a promising direction of research. In fact, coalescent embedding algorithms combine important performance improvement with a spectacular speed up both on in-silico and real tests. We hope that this article will contribute to establish a new bridge at the interface between physics of complex networks and computational machine learning theory, and that future extended studies will dig into real applications revealing the impact of coalescent network embedding in social, natural and life sciences. Interestingly, a first important instance of impact on network medicine is given in the article of Cacciola et al. [68], in which coalescent embedding in the hyperbolic space enhances our understanding of the hidden geometry of brain connectomes, introducing the idea of network latent-geometry marker characterization of brain diseases with application in de novo drug-naïve Parkinson's Disease patients.

A.4 Methods

Coalescent embedding algorithm in 2D

INPUT: adjacency matrix, x
OUTPUT: polar coordinates of nodes, (r, θ)

1. Pre-weighting rules

1.1. Repulsion-Attraction Rule (RA, local)

$$x_{ij}^{RA1} = \frac{d_i + d_j + d_i d_j}{1 + CN_{ij}} \qquad (if\ x_{ij} = 1) \tag{1}$$

$$x_{ij}^{RA2} = \frac{1 + e_i + e_j + e_i e_j}{1 + CN_{ij}} \quad (if\ x_{ij} = 1) \tag{2}$$

x_{ij} value of (i, j) entry in matrix x; d_i degree of node i; e_i external degree of node i (links neither to CN_{ij} nor to j); CN_{ij} common neighbours of nodes i and j.

1.2. Edge Betweenness Centrality Rule (EBC, global)

$$x_{ij}^{EBC} = \sum_{s,t \epsilon V} \frac{\sigma(s, t|l_{ij})}{\sigma(s, t)} \tag{3}$$

V set of nodes; s, t any combination of network nodes in V; $\sigma(s, t)$ number of shortest paths (s, t); $\sigma(s, t|l_{ij})$ number of shortest paths (s, t) through link l_{ij}.

2. Nonlinear dimension reduction

2.1 Laplacian Eigenmaps (LE), 2nd-3rd dimensions
2.2 Isomap (ISO), 1st-2nd dimensions
2.3 Noncentered Isomap (ncISO), 2nd-3rd dimensions
2.4 Minimum Curvilinearity (MCE), 1st dimension
2.5 Noncentered Minimum Curvilinearity (ncMCE), 2nd dimension

3. Angular coordinates (θ)

3.1 Circular adjustment
3.2 Equidistant adjustment

4. Radial coordinates (r)

Nodes are sorted according to descending degree and the radial coordinate of the i-th node in the set is computed according to:

$$r_i = \frac{2}{\zeta}[\beta \ln i + (1 - \beta) \ln N] \quad i = 1,2, \dots, N \tag{4}$$

N number of nodes; $\zeta = \sqrt{-K}$, we set $\zeta = 1$; K curvature of the hyperbolic space;
$\beta = \frac{1}{\gamma - 1}$ popularity fading parameter; γ exponent of power-law degree distribution

A flow chart with the visualization of the intermediate results produced by the algorithmic steps on a toy network are provided in Fig. 2.

Coalescent embedding algorithm in 3D

INPUT: adjacency matrix, x
OUTPUT: spherical coordinates of nodes, (r, θ, φ)

1. Pre-weighting rules (same as 2D)

1.1. Repulsion-Attraction Rule (RA, local)
1.2. Edge Betweenness Centrality Rule (EBC, global)

2. Nonlinear dimension reduction

2.1 Laplacian Eigenmaps (LE), 2nd-3rd-4th dimensions
2.2 Isomap (ISO), 1st-2nd-3rd dimensions
2.3 Noncentered Isomap (ncISO), 2nd-3rd-4th dimensions

3. Angular coordinates (θ, φ)

The angular coordinates obtained from the nonlinear dimension reduction are used.

4. Radial coordinates (r) (same as 2D)

Notes:
Four variants of the Repulsion-Attraction pre-weighting rule have been tested, the results are shown in Supplementary Fig. 23 and commented in Supplementary Discussion. Here only the two best rules are reported.
The exponent γ of the power-law degree distribution has been fitted using the MATLAB script 'plfit.m', according to a procedure described by Clauset et al. [91] and released at http://www.santafe.edu/~aaronc/powerlaws/.

Manifold-based embedding

The first type of topological-based unsupervised machine learning for nonlinear dimension reduction adopted in this study are Isomap, ISO [85], and Laplacian Eigenmaps LE [86]. These two methods are manifold-based machine learning because, in classical dimension reduction of multidimensional datasets, they approximate the sample data manifold using a proximity graph, and then they embed by matrix decomposition the sample distances in a two-dimensional space. In our application the proximity graph is already given, representing an important advantage, because the topological connections (similarities) between the nodes are already known. In fact, the problem to infer a proximity graph is not trivial and generally requires the introduction of at least a tuning parameter, for instance in the procedure to learn a nearest-neighbour graph (network) that approximates the manifold. Furthermore, there is not a clear strategy to unsupervisedly tune these kinds of parameters to infer the proximity graph.

ISO is based on extracting a distance matrix (or kernel) that stores all the network shortest path distances (also named geodesic distances) that approximate the real distances over the manifold. Then the kernel is centred and in this work singular value decomposition (SVD) is applied to embed the nodes in the two-dimensional space. We also propose the noncentered version of the same algorithm, named ncISO, in which the kernel centering is neglected. Consequently, the first dimension of embedding is discarded because, since it points toward the center of the manifold, is not useful. For more computational details on the implementation of ISO please refer to [34], [36], [85].

LE is a different type of manifold machine learning. In fact, the inference of a distance kernel (for instance the shortest path kernel for ISO) starting from the network structure is not required in this algorithm, which makes it faster than ISO. Indeed, the idea behind LE is to perform the eigen-decomposition of the network's Laplacian matrix, and then to perform two-dimensional embedding of the network's nodes according to the eigenvectors related to the second and third smallest eigenvalues. The first smallest eigenvalue is zero, thus the related eigenvector is neglected. In order to implement a weighted version of this algorithm we used, as suggested in the original publication [86], the 'heat-function' (instead of the pre-weighting values as they are in their original scale):

$$\tilde{x}_{ij} = e^{-\frac{{x_{ij}}^2}{t}} \tag{5}$$

Where x_{ij} is the original pre-weighing value for the link *i,j*, and t is a scaling factor fixed as the squared mean of all the network's pre-weighting values. For more computational details on the implementation of LE please refer to [86].

Minimum Curvilinearity and minimum curvilinear embedding

The centered and noncentered versions of the minimum curvilinear embedding algorithm – respectively named MCE and ncMCE - are based on a general nonlinear dissimilarity learning theory called Minimum Curvilinearity [34]. These approaches compress the information in one unique dimension because they learn nonlinear similarities by means of the minimum spanning tree, providing a hierarchical-based mapping. This is fundamentally different from the previous algorithms (Isomap and LE), which are manifold-based. If we would consider the mere unsupervised machine learning standpoint, we would notice that manifold-based techniques in this study showed two main weaknesses: 1) they offer less compression power because two orthogonal dimensions of representation, instead of one, are needed; 2) the node similarity pattern remains nonlinear (circular) also in the embedded space, thus the goal of the nonlinear dimension reduction to linearize a (hidden) nonlinear pattern in the embedding space is missed. In unsupervised tasks where the objective is to discover unknown and unexpected sample stratifications - for instance the discovery of unforeseen groups of patients with undetected molecular-based disease variations – the linearization of a pattern along one unique embedding dimension can offer an undisputed help to recognize hidden and hierarchical organized subgroups [34], [92].

However, the utility of a computational technique varies in relation to the designed target to reach. In fact, when the goal is to perform embedding of the points of a multidimensional dataset for unsupervised pattern detection and linearization, the graph that connects the points and approximates the manifold is unknown, and the minimum-curvilinearity-based techniques offer an advantage over manifold approaches. Conversely, when the task is to embed a network in the hyperbolic space, the graph that approximates the manifold is already given, and the goal of embedding is to retain - and not to linearize - the circular node similarity pattern in the two-dimensional space. Therefore the manifold-based techniques, compared to the minimum-curvilinearity-based, could offer a better approximation of the angular coordinates (representing the network node similarities) especially for high temperatures of the PSO model, when the tree-like network organization and the related hyperbolic geometry degenerate.

Interestingly, MCE and ncMCE were theoretically designed by Cannistraci et al. [34], [36] according to a previous theory of Bogugna, et al. presented in the 'Navigability of complex networks' [16]. This article was clearly explaining that to navigate efficiently a network (and thus approximate geodesic/curvilinear pairwise node connections over the hidden manifold) it was not necessary to know the complete information of network topology in the starting point of the navigation. A greedy routing process (thus, based on the neighbourhood information) was enough to efficiently navigate the network. This triggered an easy conclusion at the basis of the Minimum Curvilinearity learning theory design: to approximate curvilinear distances between the points of the manifold it is not necessary to reconstruct the nearest-neighbour graph. Just a greedy routing process (that exploits a norm, for instance Euclidean) between the points in the multidimensional space, is enough to efficiently navigate the hidden network that approximates the manifold in the multidimensional space. In few words, learning nonlinear distances over the manifold by navigating an invisible and unknown network was possible, because the navigation process was instead guided by a greedy routing. Hence, according to Cannistraci et al. [34], [36], a preferable greedy routing strategy, was the minimum spanning tree (MST). The only hypothesis of application of this approach was that the points were not homogenously distributed in a lattice regular structure, or a similar degenerative condition. Thus, the minimum curvilinear kernel is the matrix that collects all the pairwise distances between the points (or nodes) computed over the MST. And the ncMCE is the embedding of the noncentered minimum curvilinear kernel by means of the SVD. The reason why to exploit the ncMCE - noncentered version of MCE [34] - is discussed in a second article [36] that presents how to use this approach for link prediction in protein interaction networks. The main difference between MCE and ncMCE is that in general MCE linearizes the hidden patterns along the first dimension of embedding while ncMCE along the second dimension (since it is noncentered the first dimension of embedding should be generally neglected because it points towards the center of the manifold). To conclude this part, MCE/ncMCE are conceptually different from all the other approaches because they are one of the few (maybe the only, to the best of our knowledge) dimensionality reduction methods that performs hierarchical embedding, and they exploit the MST as a highway to navigate different regions of the network. Although they exploit a small fraction of the network links – practically only the MST, which consists of N-1 links in a network with N nodes - the reason why they work efficiently to infer the angular coordinates of networks that follow the PSO model is well explained in the article of Papadopoulos et al. [23] thus we take the advantage to report the full paragraph: << This work shows that random geometric graphs [93] in hyperbolic spaces are an adequate model for complex networks. The high-level explanation of this connection is that complex networks exhibit hierarchical, tree-like organization, while hyperbolic geometry is the geometry of trees [94]. Graphs representing complex networks appear then as discrete samples from the continuous world of hyperbolic geometry. >>

However, the problem to compute the MST on an unweighted adjacency matrix is that we do not have a norm that suggests the hidden connectivity geometry. Thus, there was a clear margin to improve the performance of MCE/ncMCE by pre-weighting the links in the network (and the adjacency matrix) using a convenient strategy to suggest topological distances between the connected nodes. In fact, in the Supplementary Fig. 5 and 6 we notice that the pre-weighting strategy significantly boosts MCE/ncMCE performance.

HyperMap

HyperMap [23] is a method to map a network into its hyperbolic space based on Maximum Likelihood Estimation (MLE). For sake of clarity, the first algorithm for MLE-based network embedding in the hyperbolic space is not HyperMap, but to the best of our knowledge is the algorithm proposed by Boguñá et al. in [18]. HyperMap is basically an extension of that method applied to the PSO model. Unlike the coalescent embedding techniques, it can only perform the embedding in two dimensions and cannot exploit the information of the weights. It replays the hyperbolic growth of the network and at each step it finds the polar coordinates of the added node by maximizing the likelihood that the network was produced by the E-PSO model [23]. For curvature $K = -1$ the procedure is as follows:

(1) Nodes are sorted decreasingly by degree and then labeled $i = 1, 2, \dots, N$ according to the order;

(2) Node $i = 1$ is born and assigned radial coordinate $r_1 = 0$ and a random angular coordinate $\theta_1 \in [0, 2\pi]$;

(3) For each node $i = 2, 3, \dots, N$ do:

(3.a) Node i is added to the network and assigned a radial coordinate

$$r_i = 2 \ln i \tag{6}$$

(3.b) The radial coordinate of every existing node $j < i$ is increased according to

$$r_j(i) = \beta r_j + (1 - \beta) r_i \tag{7}$$

where $\beta \in (0, 1]$ is obtained from the exponent γ of the power-law degree distribution

$$\gamma = 1 + \frac{1}{\beta} \tag{8}$$

(3.c) The node i is assigned an angular coordinate by maximizing the likelihood

$$L_i = \prod_{1 \le j < i} p(h_{ij})^{x_{ij}} (1 - p(h_{ij}))^{1 - x_{ij}} \tag{9}$$

where $p(h_{ij})$ is the connection probability of nodes i and j

$$p(h_{ij}) = \frac{1}{1 + \exp\left(\frac{h_{ij} - R_i}{2T}\right)} \tag{10}$$

which is function of the hyperbolic distance h_{ij} between node i and node j, the current radius of the hyperbolic disk R_i, and the network temperature T [23]. The maximization is done by numerically trying different angular coordinates in steps of $2\pi/N$ and choosing the one that leads to the biggest L_i. The method has been implemented in MATLAB.

HyperMap-CN

HyperMap-CN [25] is a further development of HyperMap, where the inference of the angular coordinates is not performed anymore maximizing the likelihood $L_{i,L}$,based on the connections and disconnections of the nodes, but using another local likelihood $L_{i,CN}$, based on the number of common neighbours between each node i and the previous nodes $j < i$ at final time. Here the hybrid model has been used, a variant of the method in which the likelihood $L_{i,CN}$ is only adopted for the high degree nodes and $L_{i,L}$ for the others, yielding a shorter running time. Furthermore, a speed-up heuristic and corrections steps can be applied. The speed-up can be achieved by getting an initial estimate of the angular coordinate of a node i only considering the previous nodes $j < i$ that are i's neighbours. The maximum likelihood estimation is then performed only looking at an interval around this initial estimate. Correction steps can be used at predefined times i after step 3.c (in the description of HyperMap). Each existing node $j < i$ is visited and with the knowledge of the rest of the coordinates the angle of j is updated to the value that maximizes the likelihood $L_{j,L}$. The C++ implementation of the method has been released by the authors at the website https://bitbucket.org/dk-lab/2015_code_hypermap. For the embedding of all the PSO networks the default settings (correction steps but no speed-up heuristic) have been used, whereas for all the real networks neither correction steps nor speed-up heuristic have been used.

LPCS

Link Prediction with Community Structure (LPCS) [87] is a hyperbolic embedding technique that consists of the following steps: (1) Detect the hierarchical organization of communities. (2) Order the top-level communities starting from the one that has the largest number of nodes and using the Community Intimacy index, which takes into account the proportion of edges within and between communities. (3) Recursively order the lower level communities based on the order of the higher-level communities, until reaching the bottom level in the hierarchy. (4) Assign to every bottom-level community an angular range of size proportional to the nodes in the community, in order to cover the complete circle with non-overlapping angular ranges. Sample the angular coordinates of the nodes uniformly at random within the angular range of the related bottom-level community. (5) Assign the radial coordinates according to Equation 4.

The LPCS code firstly takes advantage of the Louvain R function for detecting the hierarchy of communities (see Louvain method), then we implemented the embedding in MATLAB.

The C-score for angular coordinates evaluation

The inference of the angular coordinates order is evaluated according to a modified version of the concordance score (C-score) [95]. The C-score is defined as the proportion of sample pairs for which the ranking by a prediction model corresponds to the true ranking. Here, we need to adapt the interpretation to our particular case, where the ranked samples are not disposed along an axis but on a circle, hence the C-score can be interpreted as the proportion of node pairs for which the angular relationship in the inferred network corresponds to the angular relationship in the original network. Below, we report the formula to compute the C-score in our case:

$$C - score = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \delta(i,j)}{n * (n-1)/2} \tag{11}$$

Where: n is the total number of network nodes; i and j indicate two nodes and $\delta(i,j)$ is 1 if the shortest angular distance from i to j has the same direction (clockwise or counter-clockwise) in both the original and inferred coordinates, and 0 if the direction is the opposite in the original and inferred coordinates.

Since in the inferred network the nodes could have been arranged in the opposite clock direction with respect to the original network, the C-score is computed also considering the inferred angular relationships in the opposite clock direction (the two conditions for the value of $\delta(i,j)$ are inverted) and the maximum between the two values is chosen.

Greedy routing

An important characteristic that can be studied in a network embedded in a geometrical space is its navigability. The network is considered navigable if the greedy routing (GR) performed using the node coordinates in the geometrical space is efficient [16]. In the GR, for each pair of nodes i and j, a packet is sent from i with destination j. Each node knows only the address of its neighbours and the address of the destination j, which is written in the packet. The address of a node is represented by its coordinates in the geometrical space. In the GR procedure adopted [23], the nodes are located in the hyperbolic disk and at each hop the packet is forwarded from the current node to its neighbour closest to the destination, meaning at the lowest hyperbolic distance. The packet is dropped when this neighbour is the same from which the packet has been received at the previous hop, since a loop has been generated. In order to evaluate the efficiency of the GR, two metrics are usually taken into account: the percentage of successful paths and the average hop-length of the successful paths [16]. The first one indicates the proportion of packets that are able to reach their destinations - the higher the better - whereas the second one indicates if the successful packets require on average a short path to reach the destination - the lower the better. In order to compare the performance of methods in a unique way while taking both of the metrics into account, a GR-score has been introduced and it is computed as follows:

$$GR_{score} = \frac{\sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \frac{sp_{ij}}{p_{ij}}}{n * (n - 1)} \tag{12}$$

Where i and j are two within the set of n nodes, sp_{ij} is the shortest path length from i to j and p_{ij} is the GR path length from i to j. The ratio $\frac{sp_{ij}}{p_{ij}}$ assumes values in the interval [0, 1]. When the greedy routing is unsuccessful the path length is infinite and therefore the ratio is 0, which represents the worst case. When the greedy routing is successful the path length is greater than 0 and tends to 1 as the path length tends to the shortest path length, becoming 1 in the best case. The GR-score is the average of this ratio over all the node pairs.

Louvain algorithm for community detection

The Louvain algorithm [44] is separated into two phases, which are repeated iteratively.

At first every node in the (weighted) network represents a community in itself. In the first phase, for each node i, it considers its neighbours j and evaluates the gain in modularity that would take place by removing i from its community and placing it in the community of j. The node i is then placed in the community j for which this gain is maximum, but only if the gain is positive. If no gain is possible node i stays in its original community. This process is applied until no further improvement can be achieved.

In the second phase the algorithm builds a new network whose nodes are the communities found in the first phase, whereas the weights of the links between the new nodes are given by the sum of the weight of the links between nodes in the corresponding two communities. Links between nodes of the same community lead to self-loops for this community in the new network.

Once the new network has been built, the two phase process is iterated until there are no more changes and a maximum of modularity has been obtained. The number of iterations determines the height of the hierarchy of communities detected by the algorithm.

For each hierarchical level there is a possible partition to compare to the ground truth annotation. In this case, the hierarchical level considered is the one that guarantees the best match, therefore the detected partition that gives the highest NMI value.

We used the R function 'multilevel.community', an implementation of the method available in the 'igraph' package [96].

In this study, the embedding of the network in the hyperbolic space has been exploited in order to weight the input adjacency matrix. Given the hyperbolic coordinates, the observed links have been weighted using the formula

$$x_{ij}^{HD} = \frac{1}{1 + HD_{ij}} \quad (13)$$

where HD_{ij} is the hyperbolic distance between nodes i and j. For the Louvain algorithm a further variant has been tested in which also the non-observed links have been weighted using the formula

$$x_{ij}^{HSP} = \frac{1}{1 + HSP_{ij}} \quad (14)$$

where HSP_{ij} is the hyperbolic shortest path between nodes i and j, computed as the sum of the hyperbolic distances over the shortest path.

Infomap algorithm for community detection

The Infomap algorithm [45] finds the community structure by minimizing the expected description length of a random walker trajectory using the Huffman coding process [97].

It uses the hierarchical map equation (a further development of the map equation, to detect community structures on more than one level), which indicates the theoretical limit of how concisely a network path can be specified using a given partition structure. In order to calculate the optimal partition (community) structure, this limit can be computed for different partitions and the community annotation that gives the shortest path length is chosen.

For each hierarchical level there is a possible partition to compare to the ground truth annotation. In this case, the hierarchical level considered is the one that guarantees the best match, therefore the detected partition that gives the highest NMI value.

We used the C implementation released by the authors at http://www.mapequation.org/code.html.

In this study, the embedding of the network in the hyperbolic space has been exploited in order to weight the input adjacency matrix. Given the hyperbolic coordinates, the observed links have been weighted using the hyperbolic distances according to Equation 13.

Label propagation algorithm for community detection

The label propagation algorithm [47] initializes each node with a unique label and iteratively updates each node label with the one owned by the majority of the neighbours, with ties broken uniformly at random. The update is performed in an asynchronous way and the order of the nodes at each iteration is chosen randomly. As the labels propagate through the network, densely connected groups of nodes quickly reach a consensus on a unique label. The iterative process stops when every node has the same label as the majority its neighbours, ties included.

At the end of the procedure the nodes having the same label are grouped together to form a community. Since the aim is not the optimization of an objective function and the propagation process contains randomness, there are more possible partitions that satisfy the stop criterion and therefore the solution is not unique. For this reason the algorithm has been run for 10 independent iterations and the mean performance is reported.

We used the R function 'label.propagation.community', an implementation of the method available in the 'igraph' package [96].

In this study, the embedding of the network in the hyperbolic space has been exploited in order to weight the input adjacency matrix. Given the hyperbolic coordinates, the observed links have been weighted using the hyperbolic distances according to Equation 13.

Walktrap algorithm for community detection

The Walktrap algorithm [46] is based on an agglomerative method for hierarchical clustering: the nodes are iteratively grouped into communities exploiting the similarities between them. The nodes similarities are obtained using random walks and are based on the idea that random walks tend to get trapped into densely connected subgraphs corresponding to communities.

The agglomerative method uses heuristics to choose which communities to merge and implements an efficient way to update the distances between communities. At the end of the procedure a hierarchy of communities is obtained and each level offers a possible partition. The algorithm chooses as final result the partition that maximizes the modularity.

We used the R function 'walktrap.community', an implementation of the method available in the 'igraph' package [96].

In this study, the embedding of the network in the hyperbolic space has been exploited in order to weight the input adjacency matrix. Given the hyperbolic coordinates, the observed links have been weighted using the hyperbolic distances according to Equation 13.

Normalized Mutual Information

The evaluation of the community detection has been performed using the Normalized Mutual Information (NMI) as in [98]. The entropy can be defined as the information contained in a distribution *p(x)* in the following way:

$$H(X) = \sum_{x \in X} p(x) \log p(x) \tag{15}$$

The mutual information is the shared information between two distributions:

$$I(X,Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p_1(x)p_2(y)}\right) \tag{16}$$

To normalize the value between 0 and 1 the following formula can be applied:

$$NMI = \frac{I(X,Y)}{\sqrt{H(X)H(Y)}} \tag{17}$$

If we consider a partition of the nodes in communities as a distribution (probability of one node falling into one community), we can compute the matching between the annotation obtained by the community detection algorithm and the ground truth communities of a network as follows:

$$H(C_D) = \sum_{h=1}^{n_D} \frac{n_h^D}{N} \log(\frac{n_h^D}{N}) \tag{18}$$

$$H(C_T) = \sum_{l=1}^{n_T} \frac{n_l^T}{N} \log(\frac{n_l^T}{N}) \tag{19}$$

$$I(C_D, C_T) = \sum_{h} \sum_{l} \frac{n_{h,l}}{N} \log\left(\frac{n_{h,l}}{n_h^D n_l^T}\right) \tag{20}$$

$$NMI(C_D, C_T) = \frac{I(C_D, C_T)}{\sqrt{H(C_D)H(C_T)}} \tag{21}$$

Where:

N – number of nodes;

n^D, n^T – number of communities detected by the algorithm (*D*) or ground truth (*T*);

$n_{h,l}$ – number of nodes assigned to the *h*-th community by the algorithm and to the *l*-th community according to the ground truth annotation.

We used the MATLAB implementation available at http://commdetect.weebly.com/. As suggested in the code, when $\frac{N}{n^T} \leq 100$, the NMI should be adjusted in order to correct for chance [99].

Generation of synthetic networks using the PSO model

The Popularity-Similarity-Optimization (PSO) model [20] is a generative network model recently introduced in order to describe how random geometric graphs grow in the hyperbolic space. In this model the networks evolve optimizing a trade-off between node popularity, abstracted by the radial coordinate, and similarity, represented by the angular coordinate distance, and they exhibit many common structural and dynamical characteristics of real networks.

The model has four input parameters:

- $m > 0$, which is equal to half of the average node degree;

- $\beta \in (0, 1]$, defining the exponent $\gamma = 1 + 1/\beta$ of the power-law degree distribution;

- $T \geq 0$, which controls the network clustering; the network clustering is maximized at $T = 0$, it decreases almost linearly for $T = [0,1)$ and it becomes asymptotically zero if $T > 1$;

- $\zeta = \sqrt{-K} > 0$, where K is the curvature of the hyperbolic plane. Since changing ζ rescales the node radial coordinates and this does not affect the topological properties of networks [20], we considered $K = -1$.

Building a network of *N* nodes on the hyperbolic disk requires the following steps: (1) Initially the network is empty. (2) At time $i = 1, 2, \dots, N$ a new node *i* appears with radial coordinate as described in Equation 6 and angular coordinate θ_i uniformly sampled in $[0,2\pi]$; all the existing nodes $j < i$ increase their radial coordinates according to Equation 7 in order to simulate popularity fading. (3) If $T = 0$, the new node connects to the *m* hyperbolically closest nodes; if $T > 0$, the new node picks a randomly chosen existing node $j < i$ and, given that it is not already connected to it, it connects to it with probability $p(h_{ij})$ (see Equation 10), repeating the procedure until it becomes connected to *m* nodes. (4) The growing process stops when *N* nodes have been introduced.

Real networks dataset

The community detection methods have been tested on 8 small real networks, which represent differing systems, and on 8 large Internet networks.

The networks have been transformed into undirected, unweighted, without self-loops and only the largest connected component has been considered. The information of their ground truth communities is available. Every table, together with the results, provides also some basic statistics of the networks.

The first small network is about the Zachary's Karate Club [88], it represents the friendship between the members of a university karate club in the US. The communities are formed by a split of the club into two parts, each following one trainer.

The networks from the second to the fifth are intra-organisational networks from [90] and can be downloaded at https://toreopsahl.com/datasets/#Cross_Parker. Opsahl_8 and Opsahl_9 come from a consulting company and nodes represent employees. In Opsahl_8 employees were asked to indicate how often they have turned to a co-worker for work-related information in the past, where the answers range from: 0 - I don't know that person; 1 - Never; 2 - Seldom; 3 - Sometimes; 4 - Often; 5 - Very often. Directions were ignored. The data was turned into an unweighted network by setting a link only between employees that have at least asked for information seldom (2).

In the Opsahl_9 network, the same employees were asked to indicate how valuable the information they gained from their co-worker was. They were asked to show how strongly they agree or disagree with the following statement: "In general, this person has expertise in areas that are important in the kind of work I do." The weights in this network are also based on the following scale: 0 - Do Not Know This Person; 1 - Strongly Disagree; 2 - Disagree; 3 - Neutral; 4 - Agree; 5 - Strongly Agree. We set a link if there was an agreement (4) or strong agreement (5). Directions were ignored.

The Opsahl_10 and Opsahl_11 networks come from the research team of a manufacturing company and nodes represent employees. The annotated communities indicate the company locations (Paris, Frankfurt, Warsaw and Geneva).

For Opsahl_10 the researchers were asked to indicate the extent to which their co-workers provide them with information they use to accomplish their work. The answers were on the following scale: 0 – I do not know this person / I never met this person; 1 – Very infrequently; 2 – Infrequently; 3 – Somewhat frequently; 4 – Frequently; 5 – Very frequently. We set an undirected link when there was at least a weight of 4.

For Opsahl_11 the employees were asked about their awareness of each other's knowledge ("I understand this person's knowledge and skills. This does not necessarily mean that I have these skills and am knowledgeable in these domains, but I understand what skills this person has and domains they are knowledgeable in."). The weighting was on the scale: 0 – I do not know this person / I have never met this person; 1 – Strongly disagree; 2 – Disagree; 3 – Somewhat disagree; 4 – Somewhat agree; 5 – Agree; 6 – Strongly agree. We set a link when there was at least a 4, ignoring directions.

The Polbooks network represents frequent co-purchases of books concerning US politics on amazon.com. Ground truth communities are given by the political orientation of the books as either conservative, neutral or liberal. The network is unpublished but can be downloaded at http://www-personal.umich.edu/~mejn/netdata/, as well as with the Karate, Football and Polblogs networks.

The Football network [6] presents games between division IA colleges during regular season fall 2000. Ground truth communities are the conferences that each team belongs to.

The Polblogs [100] network consists of links between blogs about the politics in the 2004 US presidential election. The ground truth communities represent the political opinions of the blogs (right/conservative and left/liberal).

The large size networks considered for community detection are Autonomous systems (AS) Internet topologies extracted from the data collected by the Archipelago active measurement infrastructure (ARK) developed by CAIDA [101]. The connections in the topology are not physical but logical, representing AS relationships, and the annotated communities are the geographical locations (countries). ARK200909-ARK201012 are topologies collected from September 2009 to December 2010 at time steps of three months (download available at https://bitbucket.org/dk-lab/2015_code_hypermap/src/bd473d7575c35e099b520bf669d92aea81fac69b/AS_topologies/). AS201501_IPv4 is a more recent version of the IPv4 Internet topology, collected on January 2015 (download at http://www.caida.org/data/active/ipv4_routed_topology_aslinks_dataset.xml).

AS201501_IPv6 is as recent as the previous one but represents the IPv6 Internet network (download at https://www.caida.org/data/active/ipv6_allpref_topology_dataset.xml).

Complexity of coalescent embedding algorithms

The time complexity of the coalescent embedding algorithms proposed can be obtained summing up the computational cost of the main steps: pre-weighting, dimension reduction, assignment of angular coordinates and radial coordinates. All the complexities reported assume the network to be connected, therefore the number of edges E has at least the same order of complexity as the number of nodes *N*.

1) Pre-weighting rules

1.1) RA: it only requires basic operations on sparse matrices, whose complexity is proportional to the number of nonzero elements, therefore $O(E)$.

1.2) EBC: it requires the computation of the Edge Betweenness Centrality, which takes $O(EN)$ using the Brandes' algorithm for unweighted graphs [102].

2) Dimension reduction techniques

2.1) LE: the method performs the eigen-decomposition of the Laplacian matrix solving a generalized eigenvalue problem and then uses the eigenvectors related to the smallest eigenvalues, discarding the first because it is zero. The computation of all the eigenvalues of the matrix requires $O(N^3)$ [103], using the MATLAB function 'eig'. However, since the

Laplacian matrix is sparse and only a few eigenvalues are needed (*k*=3 for a 2D embedding and *k*=4 for a 3D embedding in the hyperbolic space), the MATLAB function 'eigs' can be executed. Firstly, the matrix is factorized, which requires $O(N^{3/2})$ for sparse matrices [104], then it uses the Implicitly Restarted Arnoldi Method (IRAM) [105] as implemented in ARPACK [106]. It is an iterative method whose convergence speed strictly depends on the relative gap between the eigenvalues, which makes the computational complexity difficult to analyse in terms of *N* and *E*.

2.2) ISO: the method computes as first all the pairwise shortest paths between the nodes using the Johnson's algorithm, which takes $O(EN)$ [107]. The obtained kernel is centered, which costs $O(N^2)$ due to operations on dense matrices, and finally singular value decomposition is applied, in order to obtain the singular vectors related to the largest singular values. The computation of all the singular values of the matrix requires $O(N^3)$ [103], using the MATLAB function 'svd'. However, since only a few singular values are needed (*k*=2 for a 2D embedding and *k*=3 for a 3D embedding in the hyperbolic space), the function 'lansvd' from the software package PROPACK can be exploited, which has a lower computational complexity equal to $O(kN^2)$ [108].

2.3) ncISO: the method performs the same operations as ISO, omitting the kernel centering. As a consequence, the first dimension of embedding is discarded, therefore the singular values to compute are *k*=3 for a 2D embedding and *k*=4 for a 3D embedding in the hyperbolic space.

2.4) MCE: the method computes as first a minimum spanning tree over the network using the Kruskal's algorithm, whose complexity is $O(E + X * log(N))$, where *X* is the number of edges no longer than the longest edge in the MST [109]. Starting from the minimum spanning tree, all the pairwise transversal distances between disconnected nodes are computed in order to form the MC-Kernel, the complexity of this step it $O(EN)$ with $E = (N - 1)$ hence $O(N^2)$. The following step is the kernel centering that costs $O(N^2)$. Since the angular coordinates are inferred according to the first dimension of embedding, the singular values to compute are *k*=1 for a 2D embedding in the hyperbolic space.

2.5) ncMCE: the method performs the same operations as MCE, omitting the kernel centering. As a consequence, the first dimension of embedding is discarded and the angular coordinates are inferred according to the second one. Therefore the singular values to compute are *k*=2 for a 2D embedding in the hyperbolic space.

3) Angular coordinates

3.1) Circular adjustment: the assignment requires the conversion from Cartesian to polar coordinates for the methods LE, ISO and ncISO and a rescaling of the coordinates for MCE and ncMCE, in both the cases the cost is $O(N)$.

3.2) Equidistant adjustment: a sorting operation is performed, which costs $O(NlogN)$.

4) Radial coordinates: the assignment is performed applying a given mathematical formula, which requires the nodes to be sorted by degree, the cost is therefore $O(NlogN)$.

Summarizing, if full matrix factorization techniques are used, all the methods have a complexity of $O(N^3)$. Instead, if the truncated variants are used, for ISO, ncISO, MCE and ncMCE the SVD takes $O(N^2)$ since k is a small constant, therefore they have a complexity dominated by the shortest path computation, which is $O(EN)$ in case of ISO and ncISO and $O(N^2)$ in case of MCE and ncMCE. The slowest pre-weighting is EBC and it is in the same order of complexity. The Supplementary Fig. 22 where the RA pre-weighting is used shows that LE is faster than the other methods and suggests that the computational complexity might be in the same range of RA-ncMCE or even lower, certainly not higher. Therefore we conclude that the complexity would be $O(EN)$ if the EBC pre-weighting is used; and it could be even lower, and approximatively $O(N^2)$ for LE, MCE and ncMCE, if the RA pre-weighting is used.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

For real networks data that have been obtained from publicly available sources, the corresponding URLs or references are provided in the 'Real networks dataset' section.

Code availability

The MATLAB code for performing the coalescent embedding in 2D and 3D, together with functions for the evaluation (HD-correlation, C-score, GR-score) and visualization of the embedding are publicly available at the GitHub repository:

https://github.com/biomedical-cybernetics/coalescent_embedding

Hardware and software details

Unless stated otherwise MATLAB code was used for all the methods and simulations, which were carried out on a workstation under Windows 8.1 Pro with 512 GB of RAM and 2 Intel(R) Xenon(R) CPU E5-2687W v3 processors with 3.10 GHz.

Funding

The work in C.V.C research group was mainly supported by: the independent group leader starting grant of the Technische Universität Dresden (TUD); by the Klaus Tschira Stiftung (KTS) gGmbH, Germany (Grant number: 00.285.2016); by the research application pool (Forschungspoolantrags) of TUD. S.C. PhD fellowship is supported by Lipotype GmbH. AM was partially supported by the funding provided by the Free State of Saxony in accordance with the Saxon Scholarship Program Regulation, awarded by the Studentenwerk Dresden based on the recommendation of the board of the Graduate Academy of TU Dresden. We acknowledge also support by the Centre for Information Services and High Performance Computing (ZIH) of the TUD.

Acknowledgements

We thank Alexander Mestiashvili and the BIOTEC System Administrators for their IT support, Claudia Matthes for the administrative assistance and the Centre for Information Services and High Performance Computing (ZIH) of the TUD. We are particularly grateful to Fragkiskos Papadopoulos for the valuable suggestions and for sharing the AS Internet data; to Dmitri Krioukov for the enlightening theoretical discussion on the relation between scale-freeness and hyperbolicity; as well to M. Ángeles Serrano, Marian Boguña and Maksim Kitsak for the useful discussions on this topic. We thank the MathWorks Support and Josef Schicker for the detailed information about the computational complexity of the MATLAB functions for matrix factorization. C.V.C. thanks Trey Ideker for introducing him (during a one-year visiting scholar period at UCSD in 2009) to the problem of protein-protein interactions network embedding. C.V.C. thanks Roded Sharan, Natasa Przulj and Timothy Ravasi for supporting the idea of using minimum curvilinearity for prediction of protein-protein interactions. C.V.C. thanks Franco Maria Montevecchi and Massimo Alessio for their unique mentoring during the period of PhD in which he developed minimum curvilinearity machine learning strategy. C.V.C. thanks Carlo Rovelli and Enzo Di Fabrizio for their kind feedbacks on abstract.

Author contributions

CVC envisaged the study, invented and designed the coalescent embedding, the pre-weighting and equidistant adjustment strategies, the algorithms and the experiments. CVC and AM wrote the article with input and corrections from all the other authors. GB suggested to adopt the edge-betweeness-centrality measure as a pre-weighting option. AM, JMT and SC implemented and ran the codes and performed the computational analysis with CVC help. All the authors analysed the results. AM, JMT and SC realized the figures under the CVC guidance. CVC led, directed and supervised the study.

Competing interests

The authors declare no competing financial interests.

A.5 Figures and Tables

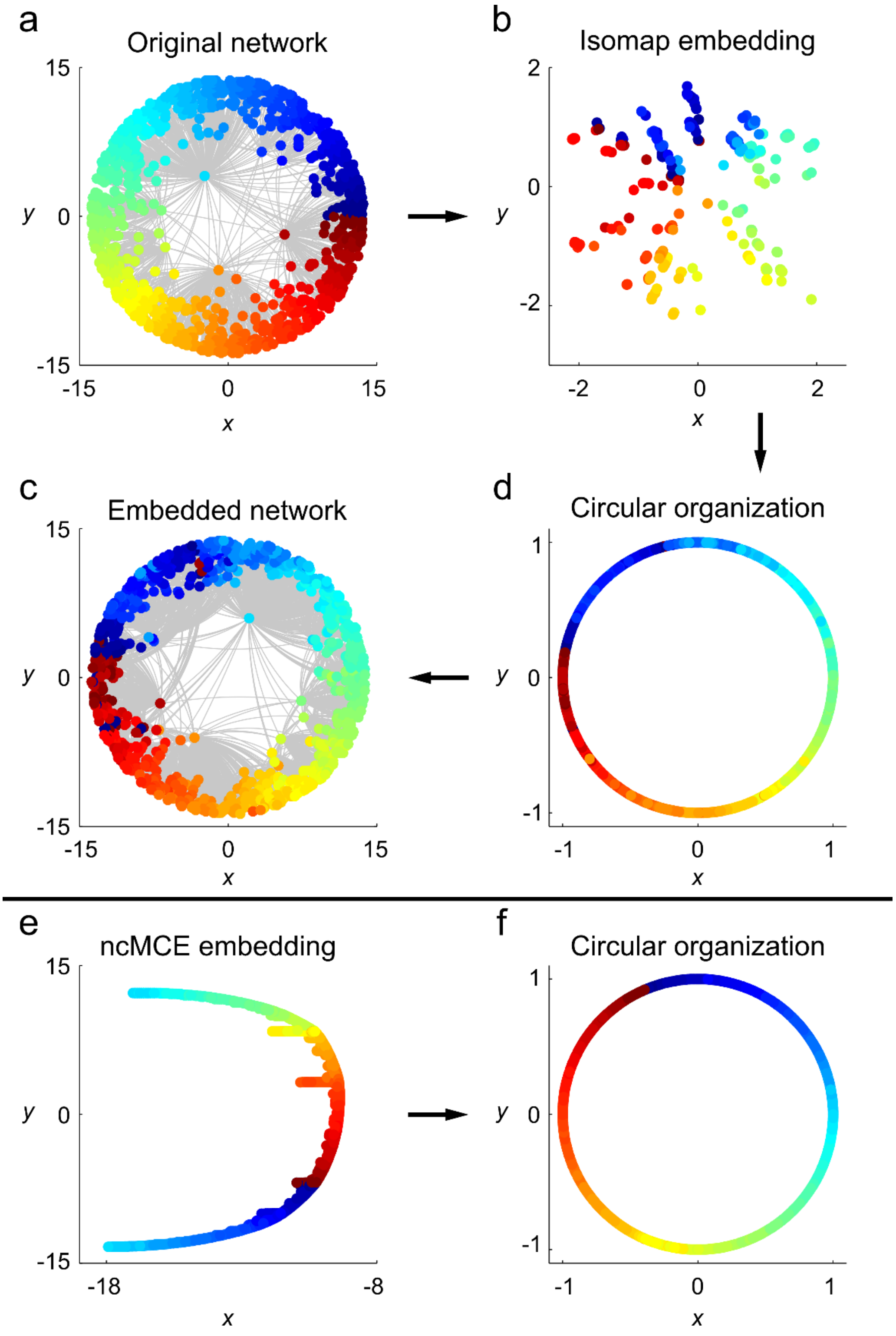


Figure 1. Coalescent embedding

(a) We show the original synthetic network generated by the PSO model in the hyperbolic space. (b) The Isomap algorithm (ISO), which is the progenitor of manifold techniques, starting from the unweighted adjacency matrix offers an embedding of the network nodes that is organized according to a circular pattern that follows the angular coordinates of the original PSO model. We made different trials using other synthetic networks, and this circular pattern is mainly preserved if the kernel is centered or if the kernel is not centered and the first dimension is neglected (see Methods for details). This makes sense because the operation of kernel centering puts the origin of the reduced space at the center of the points in a multidimensional space and thus at the center of the manifold. Since the node points lie on the hyperbolic disk, the embedding places the origin approximatively at the center of the disk. (d) The nodes are projected over a circumference and adjusted equidistantly according to the step 3.2 of the algorithm described in Methods. (c) The radial coordinates are given according to Equation 4. (e) A different pattern is obtained for an algorithm named ncMCE. The circular pattern is linearized and the nodes are ordered along the second dimension of embedding according to their similarities (here the kernel is noncentered and the first dimension of embedding should be neglected, see Methods). (f) If we accommodate the node points on the circumference following the same ordering as the second dimension of embedding, we can again recover an unbroken circular pattern that resembles the angular coordinates of the original PSO model.

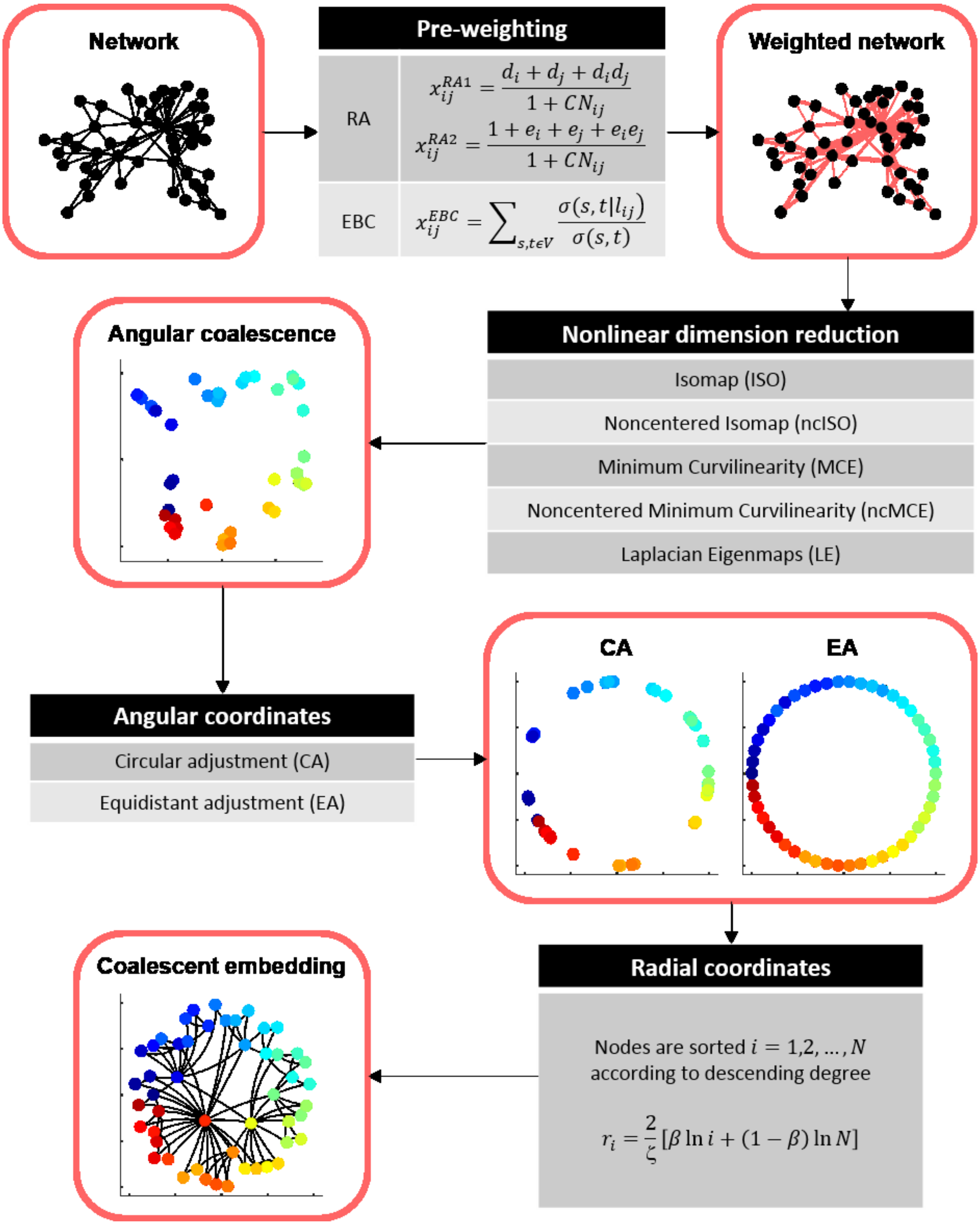

Network
Pre-weighting
RA
$x_{ij}^{RA1} = \frac{d_i + d_j + d_i d_j}{1 + CN_{ij}}$
$x_{ij}^{RA2} = \frac{1 + e_i + e_j + e_i e_j}{1 + CN_{ij}}$
EBC
$x_{ij}^{EBC} = \sum_{s,t \epsilon V} \frac{\sigma(s,t|l_{ij})}{\sigma(s,t)}$
Weighted network
Nonlinear dimension reduction
Isomap (ISO)
Noncentered Isomap (ncISO)
Minimum Curvilinearity (MCE)
Noncentered Minimum Curvilinearity (ncMCE)
Laplacian Eigenmaps (LE)
Angular coalescence
Angular coordinates
Circular adjustment (CA)
Equidistant adjustment (EA)
CA
EA
Radial coordinates
Nodes are sorted $i = 1,2, \ldots, N$ according to descending degree
$r_i = \frac{2}{\zeta}[\beta \ln i + (1 - \beta) \ln N]$
Coalescent embedding

Figure 2. Flow chart of the coalescent embedding algorithm.

The algorithmic steps (greyscale squares) and the intermediate input/output (rounded red squares) of the coalescent embedding algorithm are illustrated. Each algorithmic step reports all the possible variants. The example network has been generated by the PSO model with parameters $N = 50$, $m = 2$, $T = 0.1$, $\gamma = 2.5$. We applied the RA_1 pre-weighting rule and the ISO dimension reduction technique. The colours of the embedded nodes are assigned according to their angular coordinates in the original PSO network. Description of the variables in the mathematical formulas: x_{ij} value of (i,j) link in adjacency matrix x; d_i degree of node i; e_i external degree of node i (links neither to CN_{ij} nor to j); CN_{ij} common neighbours of nodes i and j; V set of nodes; s,t any combination of network nodes in V; $\sigma(s,t)$ number of shortest paths (s,t); $\sigma(s,t|l_{ij})$ number of shortest paths (s,t) through link l_{ij}; N number of nodes; $\zeta = \sqrt{-K}$, we set $\zeta = 1$; K curvature of the hyperbolic space; $\beta = \frac{1}{\gamma-1}$ popularity fading parameter; γ exponent of power-law degree distribution. Details on each step are provided in the respective Methods sections.

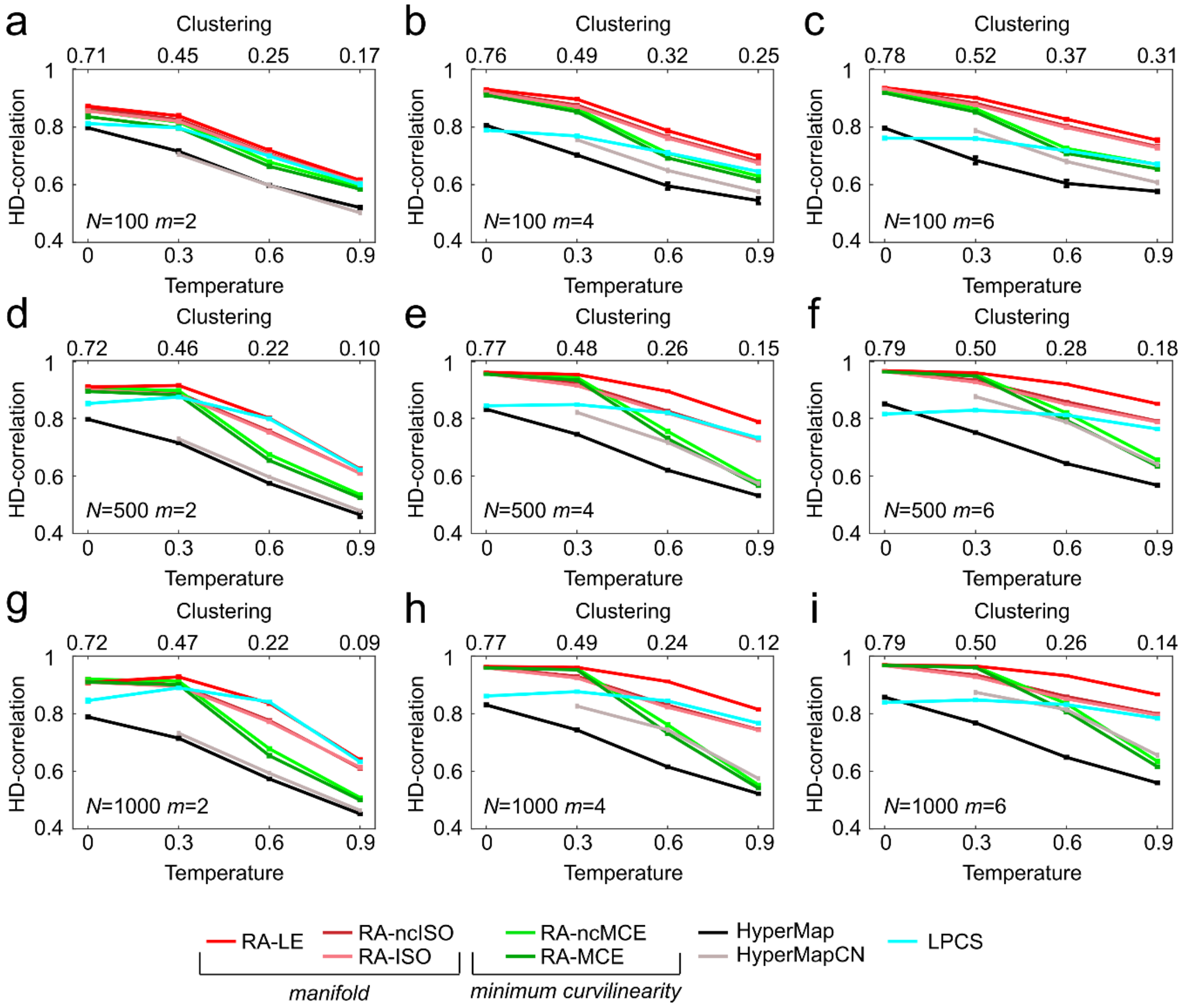


Figure 3. HD-correlation on PSO synthetic networks.

(a-i) To validate the abovementioned techniques, we generated 100 different synthetic networks for each combination of tuneable parameters of the PSO model (temperature T, size N, half of average degree m, power-law degree distribution exponent γ). Supplementary Fig. 24 offers an idea of the topological diversity of the synthetic networks generated fixing $\gamma = 2.5$ and tuning the other parameters, Supplementary Fig. 25 reports an analysis of the rich-clubness of the networks, commented in Supplementary Discussion. In the results presented in the figures of this article we used $\gamma = 2.5$, but we also ran the simulations for $\gamma = 2.25$ and 2.75, and the differences were negligible (results not shown). Here, the performance was evaluated as Pearson correlation between all the pairwise hyperbolic distances of the network nodes in the original PSO model and in the reconstructed hyperbolic space (HD-correlation). The plots report the average correlation and the standard error over the 100 synthetic networks that have been generated for each different parameter combination. The value one indicates a perfect correlation between the nodes' hyperbolic distances in the original and reconstructed hyperbolic space. The plots show the results of different methods when both RA and EA are applied. The methods without EA are plotted in Supplementary Fig. 7. For each subplot the value of HyperMap-CN for $T = 0$ is missing because the original code assumes $T > 0$.

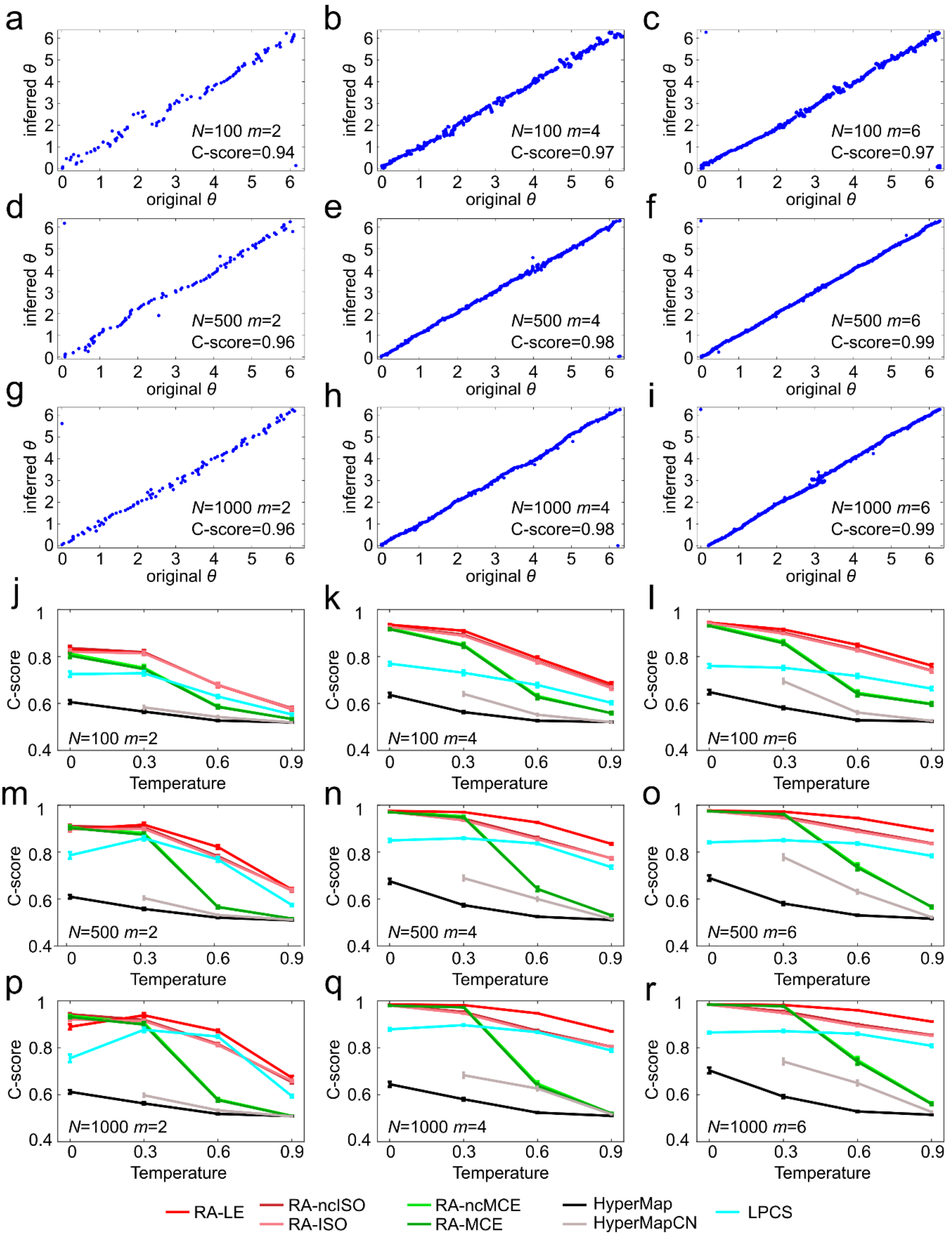

a
b
c
inferred θ
original θ
N=100 m=2
C-score=0.94
N=100 m=4
C-score=0.97
N=100 m=6
C-score=0.97
d
e
f
N=500 m=2
C-score=0.96
N=500 m=4
C-score=0.98
N=500 m=6
C-score=0.99
g
h
i
N=1000 m=2
C-score=0.96
N=1000 m=4
C-score=0.98
N=1000 m=6
C-score=0.99
j
k
l
m
n
o
p
q
r
C-score
Temperature
N=100 m=2
N=100 m=4
N=100 m=6
N=500 m=2
N=500 m=4
N=500 m=6
N=1000 m=2
N=1000 m=4
N=1000 m=6
RA-LE
RA-ncISO
RA-ISO
RA-ncMCE
RA-MCE
HyperMap
HyperMapCN
LPCS

Figure 4. Angular coordinates comparison and C-scores on PSO synthetic networks
(a-i) For all the combinations of the PSO parameters N (size) and m (half of average degree), we chose among the synthetic networks embedded with RA-MCE-EA the ones with the best C-score, which had always temperature $T = 0$. For these networks we plotted the aligned inferred angular coordinates against the original angular coordinates (θ). The alignment was done in the following manner: we applied 360 rotations of one degree both to the inferred coordinates as they are and to the inferred coordinates obtained arranging the nodes in the opposite clock direction. Then from these resulting 720 alternatives of the inferred angular coordinates we chose the one that maximizes the correlation with the original angular coordinates, in order to guarantee the best alignment. The alignment does not change the C-score, which represents the percentage of node pairs in the same circular order in the original and inferred networks (see Methods for details). Similar plots for the other coalescent embedding methods and temperature values can be found in Supplementary Fig. 8-17. (j-r) The plots report the average C-score and the standard error over the 100 synthetic networks that have been generated for each different parameter combination. There are no separate plots for the methods with and without EA since this adjustment affects the distances but not the circular ordering, therefore it does not change the C-score. For each subplot the value of HyperMap-CN for $T = 0$ is missing because the original code assumes $T > 0$.

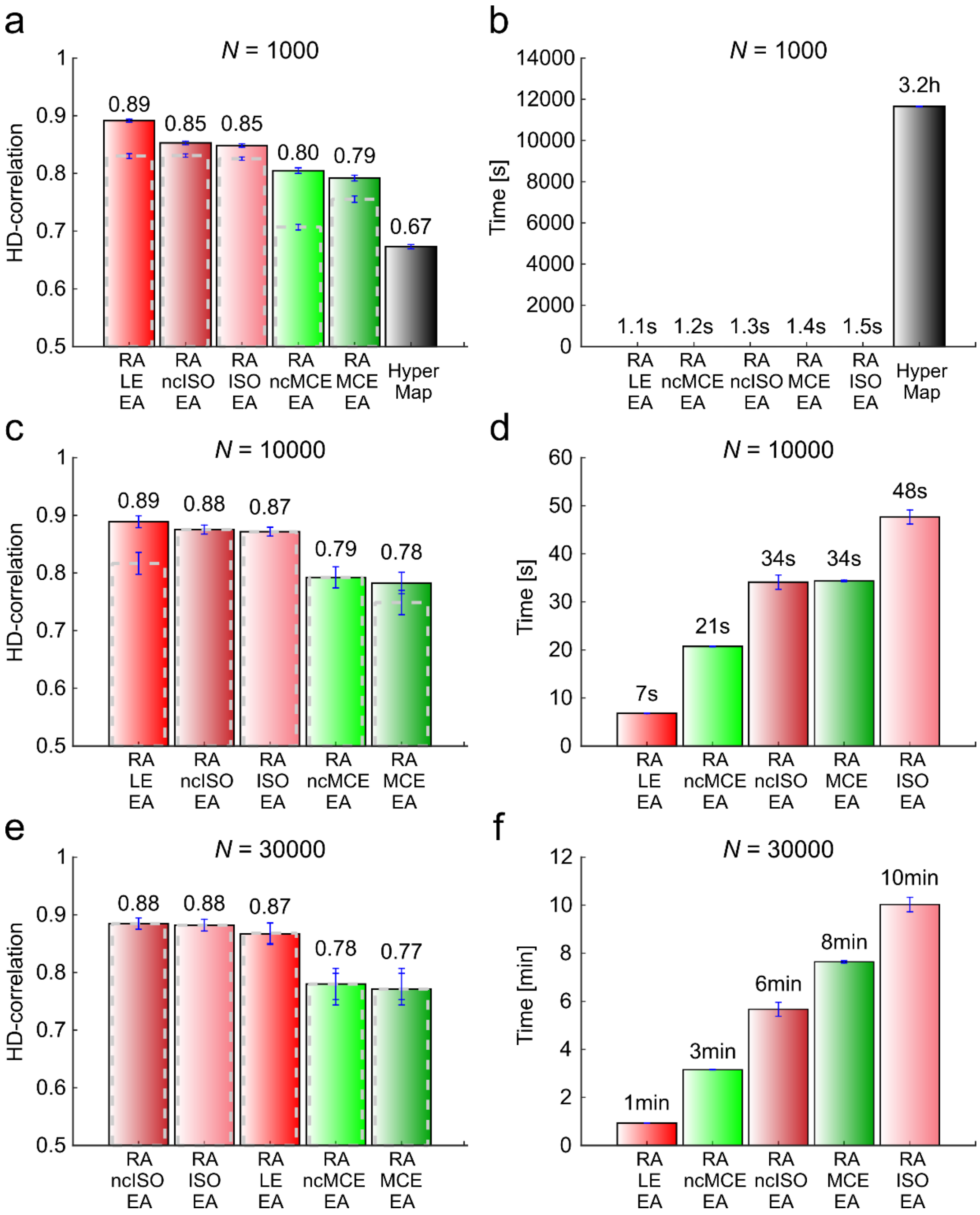

a
N = 1000
HD-correlation
1
0.9
0.8
0.7
0.6
0.5
0.89
0.85
0.85
0.80
0.79
0.67
RA LE EA
RA ncISO EA
RA ISO EA
RA ncMCE EA
RA MCE EA
Hyper Map
b
N = 1000
Time [s]
14000
12000
10000
8000
6000
4000
2000
0
1.1s
1.2s
1.3s
1.4s
1.5s
3.2h
RA LE EA
RA ncMCE EA
RA ncISO EA
RA MCE EA
RA ISO EA
Hyper Map
c
N = 10000
HD-correlation
1
0.9
0.8
0.7
0.6
0.5
0.89
0.88
0.87
0.79
0.78
RA LE EA
RA ncISO EA
RA ISO EA
RA ncMCE EA
RA MCE EA
d
N = 10000
Time [s]
60
50
40
30
20
10
0
7s
21s
34s
34s
48s
RA LE EA
RA ncMCE EA
RA ncISO EA
RA MCE EA
RA ISO EA
e
N = 30000
HD-correlation
1
0.9
0.8
0.7
0.6
0.5
0.88
0.88
0.87
0.78
0.77
RA ncISO EA
RA ISO EA
RA LE EA
RA ncMCE EA
RA MCE EA
f
N = 30000
Time [min]
12
10
8
6
4
2
0
1min
3min
6min
8min
10min
RA LE EA
RA ncMCE EA
RA ncISO EA
RA MCE EA
RA ISO EA

Figure. 5. Comparison of HD-correlation and time on small and large size PSO synthetic networks (a, c, e) Average performance and standard error, measured as HD-correlation, for all PSO networks of sizes $N = 1000$, $N = 10000$ and $N = 30000$ respectively. Averages are taken over the parameters m (half of the mean node degree) and temperature T. (b, d, f) Average computation times for the PSO networks of sizes $N = 1000$, $N = 10000$ and $N = 30000$ respectively. Again, averages are taken over the parameters m (half of the mean node degree) and temperature T. Considering the average performance in all the simulations on 1000 nodes networks (a), coalescent embedding approaches achieved a performance improvement of more than 30% in comparison to HyperMap, requiring only around one second versus more than three hours of computation time. Similar performance results are confirmed for the networks of sizes $N = 10000$ and $N = 30000$ with an execution time still in the order of minutes for the biggest networks. The comparison to HyperMap was not possible due to its long running time. The dashed grey bins represent the HD-correlation of the respective non-EA variants, suggesting that their performance tends to the EA variants for larger PSO networks.

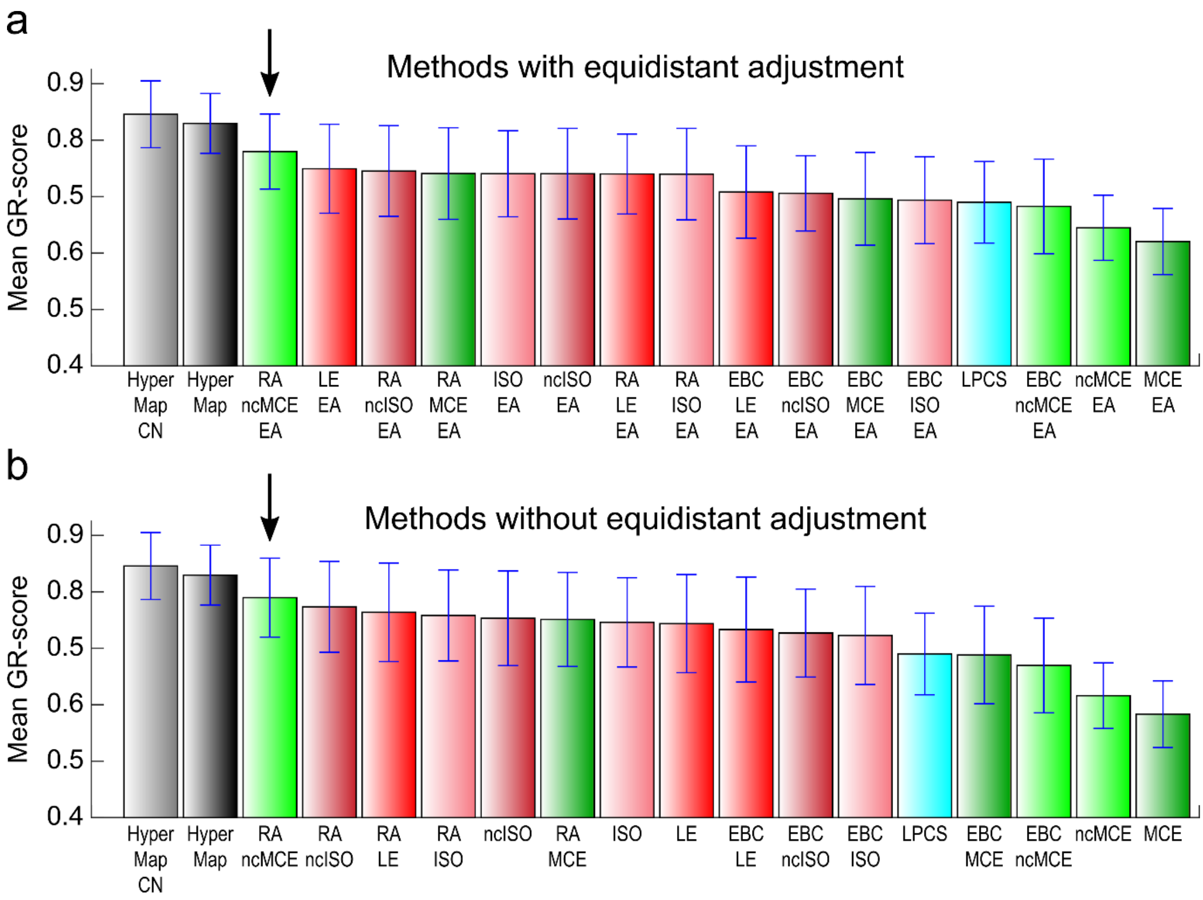


Figure. 6. Greedy routing on real networks

The 8 real networks whose statistics are reported in Table 1 have been mapped using the hyperbolic embedding techniques and the greedy routing in the geometrical space has been evaluated. The barplot report for each method the mean GR-score and standard error over the networks. The GR-score is a metric to evaluate the efficiency of the greedy routing, which assumes values between 0, when all the routings are unsuccessful, and 1, when all the packets reach the destination through the shortest path (see Methods for details). Both the EA (a) and non-EA (b) variants are reported, in order to check whether the equidistant adjustment might affect the navigability. A black arrow points the coalescent embedding algorithm RA-ncMCE that offers the best performance regardless the use of node angular adjustment. The mean GR-score of RA-ncMCE is not statistically different from the one of the HyperMap-based algorithms (permutation test p-value > 0.2 in all the pairwise comparisons).

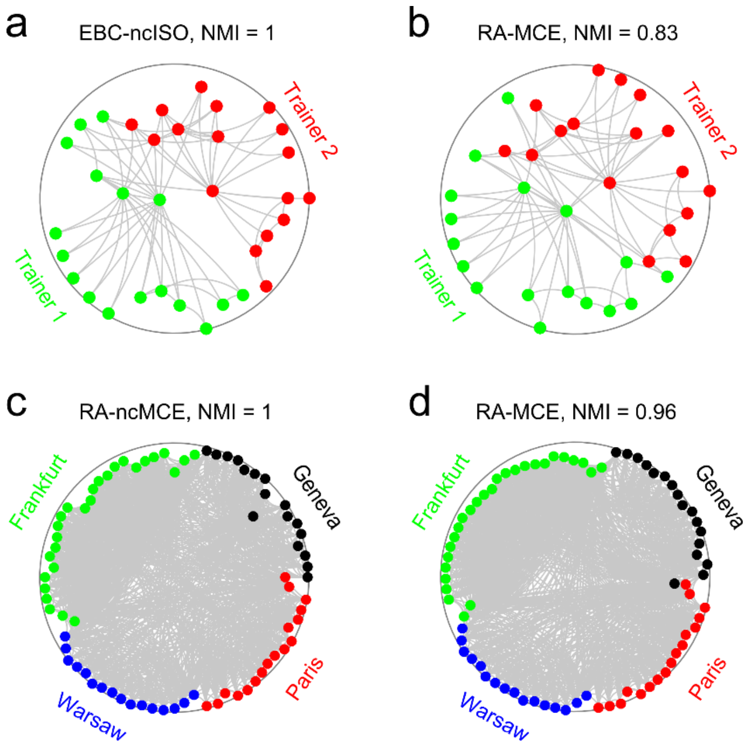


Figure. 7. Communities in Karate and Opsahl_11 networks

(a, b) Karate network embedded with EBC-ncISO-EA and RA-MCE-EA. The network represents the friendship between the members of a university karate club in the US. The two real communities are highlighted, they are formed by a split of the club into two parts, each following one trainer. The NMI obtained by the Louvain community detection algorithm is reported, where the embedding coordinates were used to weight the input matrix: observed links are weighted using the hyperbolic distances between the nodes and non-observed links using the hyperbolic shortest paths (see Methods for details). NMI is the normalized mutual information and represents the shared information between two distributions, normalized between 0 and 1, where 1 indicates that the communities detected by the algorithm perfectly correspond to the ground truth communities (see Methods for details). (c, d) Opsahl_11 network embedded with RA-ncMCE-EA and RA-MCE-EA. This is a type of intra-organisational network where a link indicates that the connected employees have both awareness of each other's knowledge and skills on the job. The four real communities are highlighted, they are related with the location of the employers. All the approaches here adopted are adjusted according to EA strategy, although this is not explicitly reported in the subtitles for brevity. Note that the angular coordinates of the embedding in (b) and (d) have been aligned for a better visualization respectively to the ones in (a) and (c), as described for the scatterplots in Fig. 4A.

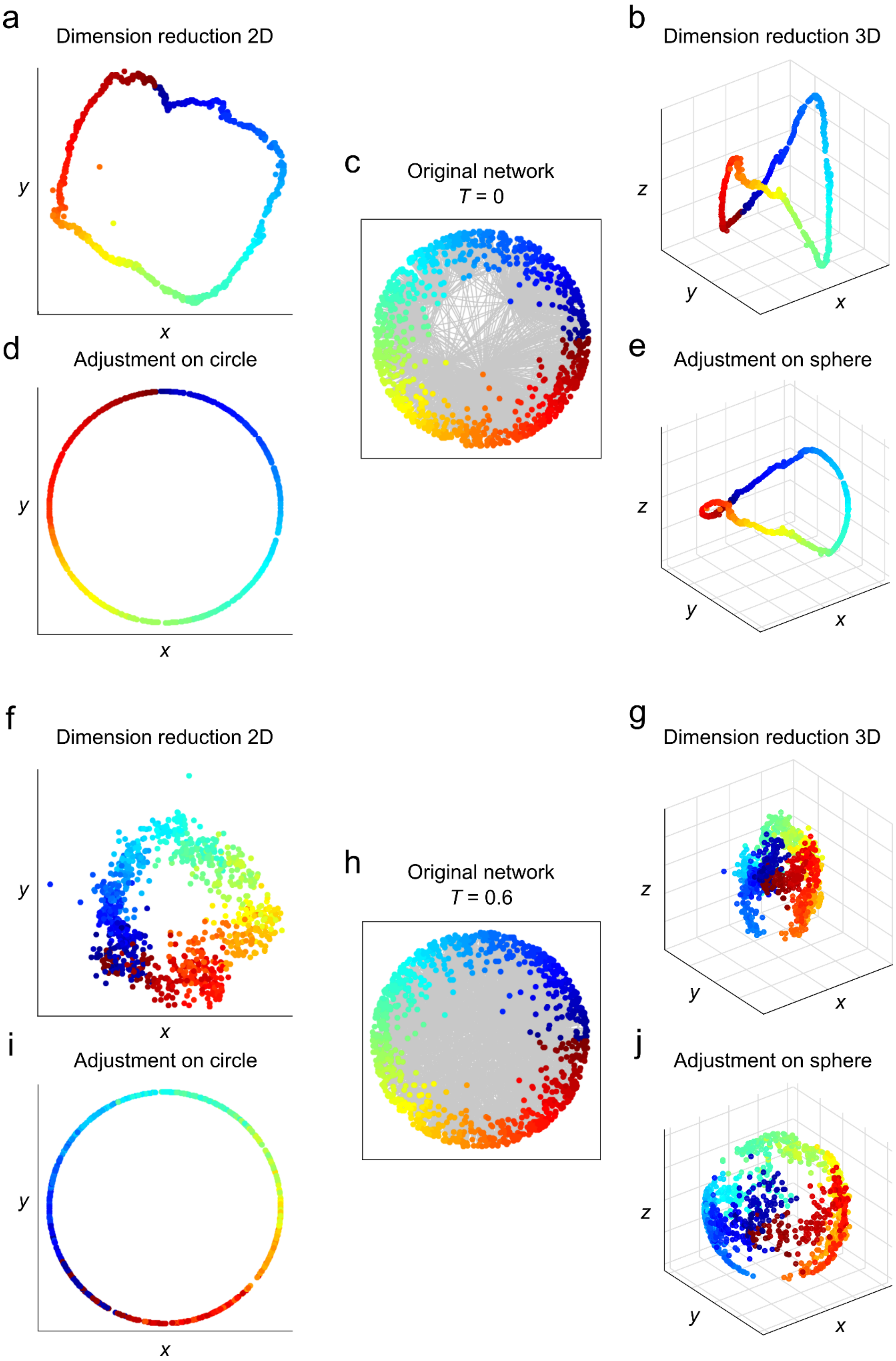

a
Dimension reduction 2D
y
x
b
Dimension reduction 3D
z
y
x
c
Original network
$T = 0$
d
Adjustment on circle
y
x
e
Adjustment on sphere
z
y
x
f
Dimension reduction 2D
y
x
g
Dimension reduction 3D
z
y
x
h
Original network
$T = 0.6$
i
Adjustment on circle
y
x
j
Adjustment on sphere
z
y
x

Figure. 8. Comparison of 2D and 3D RA-ISO embedding for increasing temperature

The figure shows how the similarities of the original PSO network ($N = 1000$, $m = 6$, $\gamma = 2.5$) (c, h) are recovered either embedding in 2D (a, f) and arranging the angular coordinates over the circumference of a circle (d, i) or embedding in 3D (b, g) and adjusting the angular coordinates over a sphere (e, j). (a-e) At low temperature (T = 0) the nodes appear distributed over a well-defined closed 3D curve. Intuitively, it seems that the 2D hyperbolic disk already offers a perfect discrimination of the similarities and with the addition of the third dimension there is not much gain of information. (f-j) With the increase of the temperature (T = 0.6 reported, T = 0.3 and T = 0.9 shown in Supplementary Fig. 21) the nodes look more and more spread around the closed 3D curve that was well-defined at low temperature. Even if one angular dimension of the sphere still recovers most of the similarities present in the original network, it is unknown if the higher spread along the second angular dimension consists in a more refined discrimination between similar nodes or in noise, therefore we have evaluated the improvement given by the 3D mapping for the greedy routing and the community detection applications.

Method	Karate	Opsahl 8	Opsahl 9	Opsahl 10	Opsahl 11	Polbooks	Football	Polblogs	Mean	% Impr.
	N=34	N=43	N=44	N=77	N=77	N=105	N=115	N=1222		
	E=78	E=193	E=348	E=518	E=1088	E=441	E=613	E=16714		
	m=2.29	m=4.49	m=7.91	m=6.73	m=14.13	m=4.20	m=5.33	m=13.68		
	T=0.43	T=0.43	T=0.32	T=0.35	T=0.28	T=0.51	T=0.60	T=0.68		
	γ=2.12	γ=8.20	γ=5.92	γ=5.06	γ=4.87	γ=2.62	γ=9.09	γ=2.38		
	N_c=2	N_c=7	N_c=7	N_c=4	N_c=4	N_c=3	N_c=12	N_c=2		
EBC-ncISO-EA	**1.00**	0.57	0.47	1.00	0.93	0.59	0.90	0.68	0.77	**+13.2**
RA-MCE-EA	**0.83**	0.51	0.47	1.00	**0.96**	0.57	0.82	0.67	0.73	+7.4
RA-ncMCE-EA	0.73	0.55	0.47	1.00	**1.00**	0.57	0.83	0.67	0.73	+7.4
EBC-MCE-EA	0.83	0.47	0.41	1.00	0.96	0.57	0.90	0.62	0.72	+5.9
EBC-ncMCE-EA	0.88	0.46	0.41	1.00	0.96	0.57	0.85	0.62	0.72	+5.9
EBC-ISO-EA	0.83	0.42	0.47	1.00	0.89	0.59	0.88	0.66	0.72	+5.9
LPCS	0.83	0.49	0.41	1.00	0.96	0.55	0.87	0.67	0.72	+5.9
ncMCE-EA	0.73	0.47	0.47	1.00	0.96	0.57	0.89	0.62	0.71	+4.4
RA-LE-EA	0.67	0.48	0.53	1.00	0.92	0.56	0.82	0.70	0.71	+4.4
RA-ncISO-EA	0.67	0.54	0.42	1.00	0.92	0.56	0.86	0.67	0.70	+2.9
ncISO-EA	0.73	0.50	0.41	1.00	0.88	0.54	0.87	0.66	0.70	+2.9
EBC-LE-EA	0.85	0.42	0.41	0.96	0.92	0.56	0.85	0.62	0.70	+2.9
MCE-EA	0.64	0.47	0.47	0.96	0.92	0.55	0.86	0.62	0.69	+1.5
unweighted	**0.46**	0.55	0.41	1.00	**0.96**	0.50	0.93	0.64	0.68	**0.0**
LE-EA	0.63	0.55	0.41	1.00	0.78	0.55	0.82	0.67	0.68	0.0
RA-ISO-EA	0.57	0.43	0.44	1.00	0.88	0.54	0.86	0.67	0.67	-1.5
ISO-EA	0.34	0.50	0.41	0.96	0.93	0.56	0.82	0.67	0.65	-4.4
HyperMap	0.56	0.60	0.28	0.92	0.85	0.50	0.83	0.69	0.65	-4.4
HyperMapCN	0.55	0.47	0.41	0.93	0.79	0.54	0.79	0.70	0.65	-4.4

Table 1. Community detection on real networks with Louvain algorithm

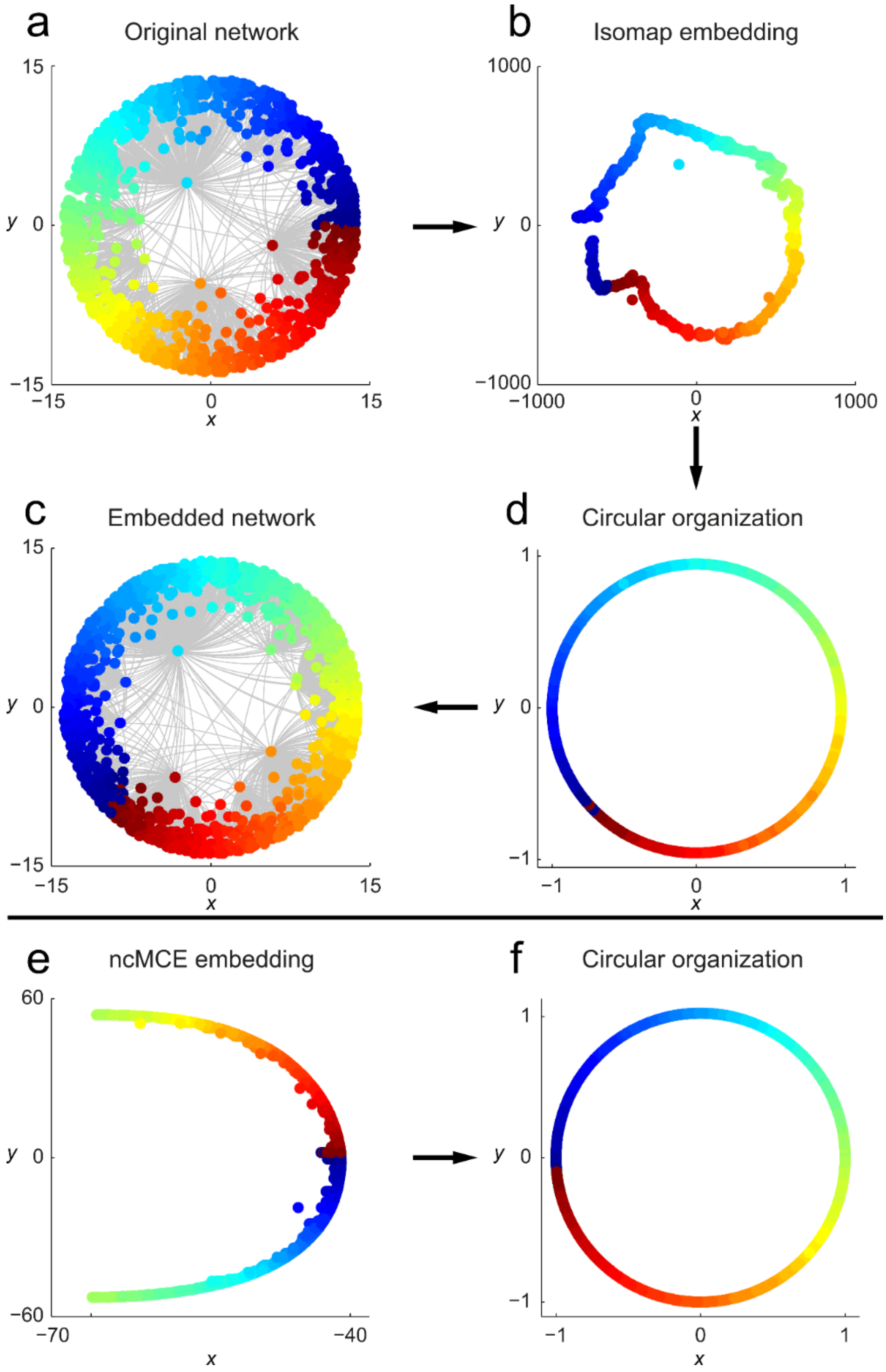
Normalized Mutual Information (NMI) computed between the ground truth communities and the ones detected by the Louvain algorithm for 8 real networks. NMI = 1 indicates a perfect match between the two partitions of the nodes. For each method, the network has been embedded in the hyperbolic space and the embedding coordinates are used to weight the input matrix for the Louvain algorithm: observed links are weighted using the hyperbolic distances between the nodes and non-observed links using the hyperbolic shortest paths (see Methods for details). As a reference, the Louvain algorithm has been run giving in input also the unweighted adjacency matrix, the related row is highlighted. The table contains also some statistics for each network: number of nodes N, number of edges E, temperature T (inversely related to the clustering coefficient), power-law degree distribution exponent γ, half of average degree m and number of ground truth communities N_c. Due to the higher performance only the EA methods are here reported, whereas the complete table is shown as Supplementary Table 1. The NMI values highlighted for the Karate and Opsahl_11 networks are the ones whose embedding is shown in Fig. 7. The rightmost column reports the percentage of improvement with respect to the unweighted variant, the best result is highlighted. The results obtained only weighting the observed links are shown in Supplementary Table 5.

Method	Karate	Opsahl 8	Opsahl 9	Opsahl 10	Opsahl 11	Polbooks	Football	Polblogs	Mean	% Impr.
	N=34	N=43	N=44	N=77	N=77	N=105	N=115	N=1222		
	E=78	E=193	E=348	E=518	E=1088	E=441	E=613	E=16714		
	m=2.29	m=4.49	m=7.91	m=6.73	m=14.13	m=4.20	m=5.33	m=13.68		
	T=0.43	T=0.43	T=0.32	T=0.35	T=0.28	T=0.51	T=0.60	T=0.68		
	γ=2.12	γ=8.20	γ=5.92	γ=5.06	γ=4.87	γ=2.62	γ=9.09	γ=2.38		
	N_c=2	N_c=7	N_c=7	N_c=4	N_c=4	N_c=3	N_c=12	N_c=2		
EBC-ncISO-EA	0.68	0.75	0.47	1.00	1.00	0.54	0.92	0.53	0.74	+4.2
ncMCE-EA	0.68	0.74	0.47	1.00	0.93	0.50	0.92	0.52	0.72	+1.4
unweighted	0.55	0.69	0.47	1.00	1.00	0.52	0.92	0.52	0.71	0.0
EBC-MCE-EA	0.68	0.55	0.53	1.00	0.96	0.52	0.93	0.52	0.71	0.0
EBC-ncMCE-EA	0.58	0.55	0.53	1.00	1.00	0.52	0.93	0.52	0.70	-1.4
ISO-EA	0.68	0.53	0.47	1.00	0.96	0.52	0.92	0.53	0.70	-1.4
LE-EA	0.68	0.54	0.47	1.00	0.96	0.53	0.92	0.51	0.70	-1.4
EBC-LE-EA	0.68	0.55	0.47	0.95	0.96	0.52	0.93	0.53	0.70	-1.4
ncISO-EA	0.68	0.53	0.47	1.00	0.96	0.47	0.92	0.53	0.69	-2.8
EBC-ISO-EA	0.55	0.55	0.47	1.00	1.00	0.52	0.92	0.53	0.69	-2.8
MCE-EA	0.68	0.54	0.47	0.95	0.93	0.51	0.92	0.52	0.69	-2.8
RA-ncISO-EA	0.55	0.55	0.47	1.00	1.00	0.52	0.92	0.52	0.69	-2.8
RA-ISO-EA	0.58	0.55	0.47	1.00	0.96	0.52	0.92	0.52	0.69	-2.8
RA-ncMCE-EA	0.47	0.55	0.53	1.00	1.00	0.52	0.92	0.50	0.69	-2.8
LPCS	0.55	0.55	0.53	1.00	0.96	0.52	0.93	0.51	0.69	-2.8
RA-LE-EA	0.55	0.55	0.47	1.00	0.93	0.52	0.92	0.52	0.68	-4.2
RA-MCE-EA	0.47	0.55	0.53	1.00	0.92	0.52	0.92	0.51	0.68	-4.2
HyperMapCN	0.52	0.55	0.41	1.00	0.86	0.57	0.89	0.46	0.66	-7.0
HyperMap	0.52	0.60	0.32	1.00	0.92	0.49	0.90	0.46	0.65	-8.5

Table 2. Community detection on real networks with Infomap algorithm

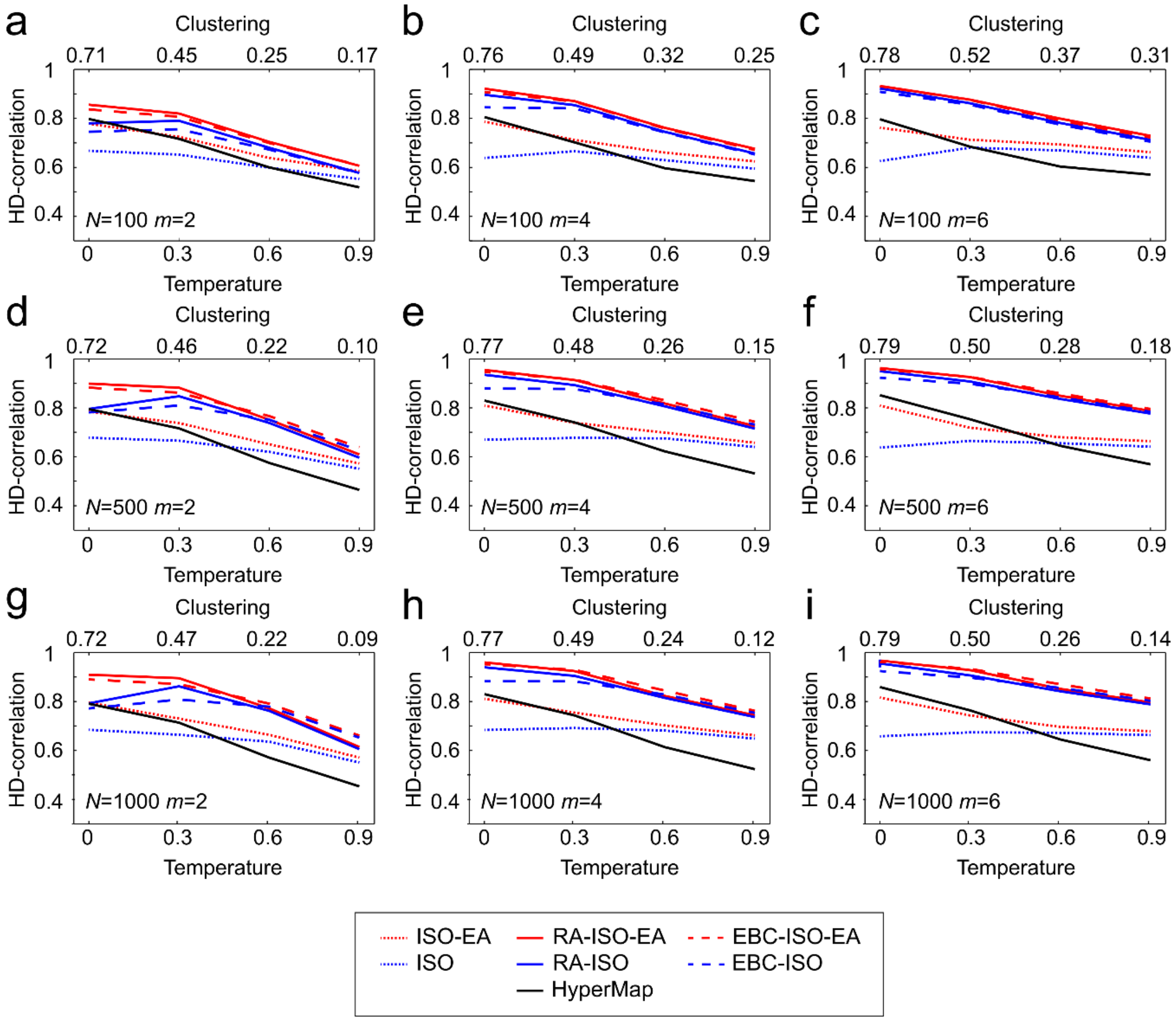
Normalized Mutual Information (NMI) computed between the ground truth communities and the ones detected by the Infomap algorithm for 8 real networks. NMI = 1 indicates a perfect match between the two partitions of the nodes. For each method, the network has been embedded in the hyperbolic space and the hyperbolic distances between the nodes are used to weight the observed links in the input matrix for the Infomap algorithm (see Methods for details). As a reference, the Infomap algorithm has been run giving in input also the unweighted adjacency matrix, the related row is highlighted. The table contains also some statistics for each network: number of nodes N, number of edges E, temperature T (inversely related to the clustering coefficient), power-law degree distribution exponent γ, half of average degree m and number of ground truth communities N_c. Due to the higher performance only the EA methods are here reported, whereas the complete table is shown as Supplementary Table 2. The rightmost column reports the percentage of improvement with respect to the unweighted variant.

A.7 Supplementary Information



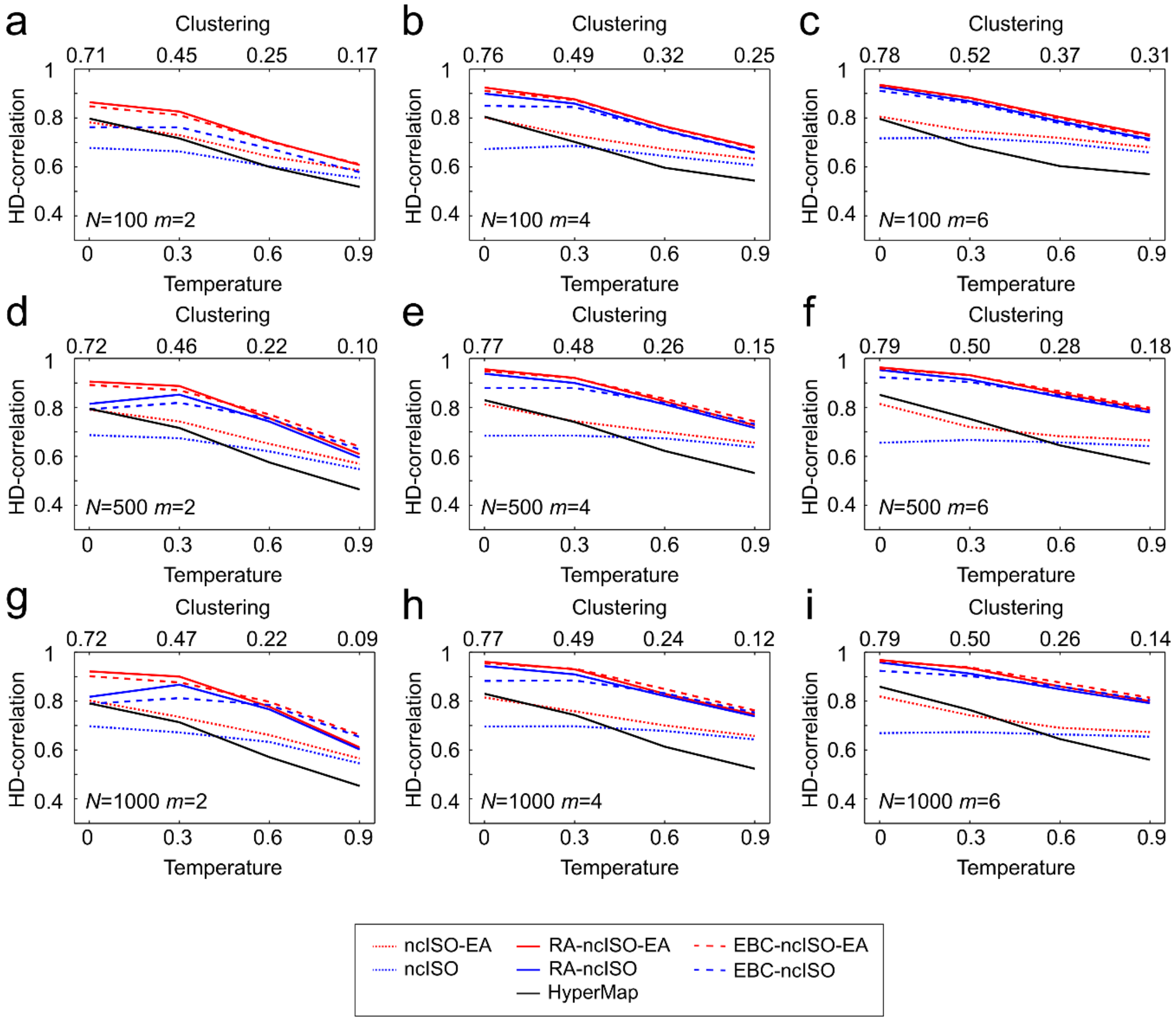
Supplementary Figure 1. Coalescent embedding with RA pre-weighting

(a) We show the original synthetic network generated by the PSO model in the hyperbolic space. (b) The Isomap algorithm (ISO) starting from the adjacency matrix pre-weighted with the repulsion-attraction (RA) rule offers an embedding of the network nodes that is organized according to a circular pattern that follows the angular coordinates of the original PSO model. The circular pattern is visible more clearly compared to the embedding without the pre-weighting (Fig. 1). (d) The nodes are projected over a circumference and adjusted equidistantly according to the step 3.2 of the algorithm described in Methods. (c) The radial coordinates are given according to Equation 4. (e) A different pattern is obtained for an algorithm named ncMCE. The circular pattern is linearized and the nodes are ordered along the second dimension of embedding according to their similarities. (f) If we accommodate the node points on the circumference following the same ordering as the second dimension of embedding, we can again recover an unbroken circular pattern that resembles the angular coordinates of the original PSO model.

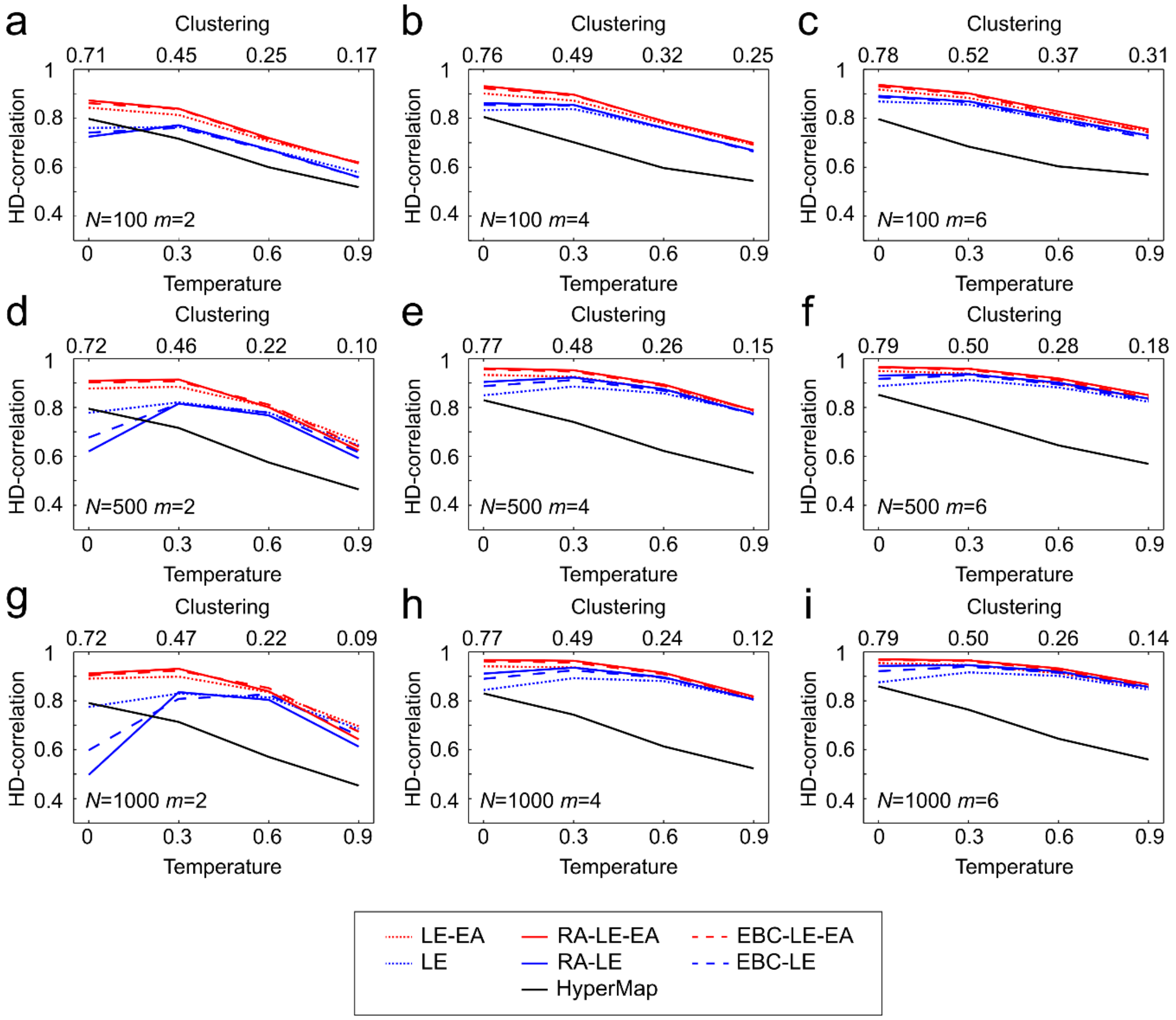


Supplementary Figure 2. HD-correlation on PSO synthetic networks for ISO

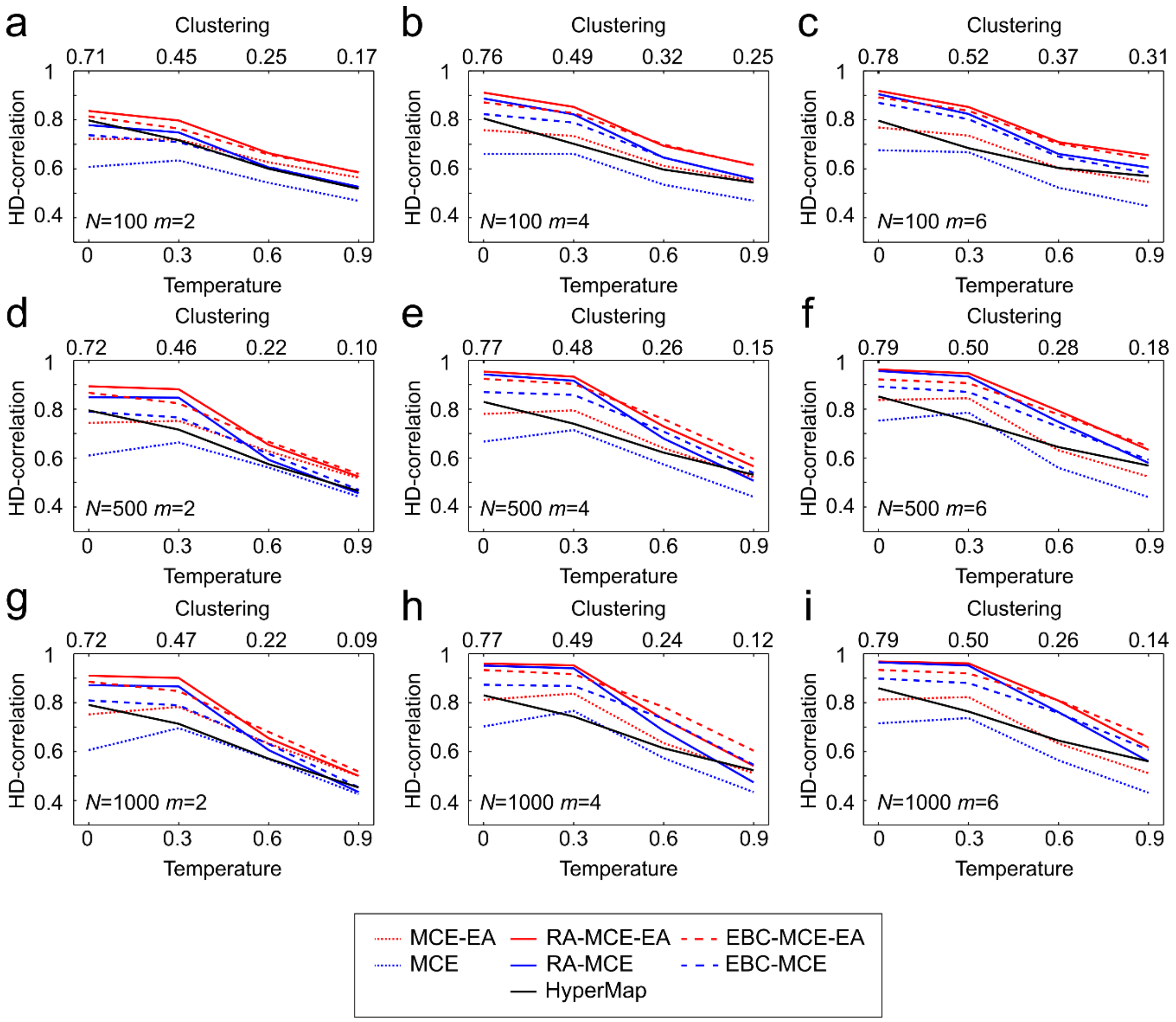
(a-i) The figure is equivalent to Fig. 3, but all the variants of ISO are compared.



Supplementary Figure 3. HD-correlation on PSO synthetic networks for ncISO
(a-i) The figure is equivalent to Fig. 3, but all the variants of ncISO are compared.

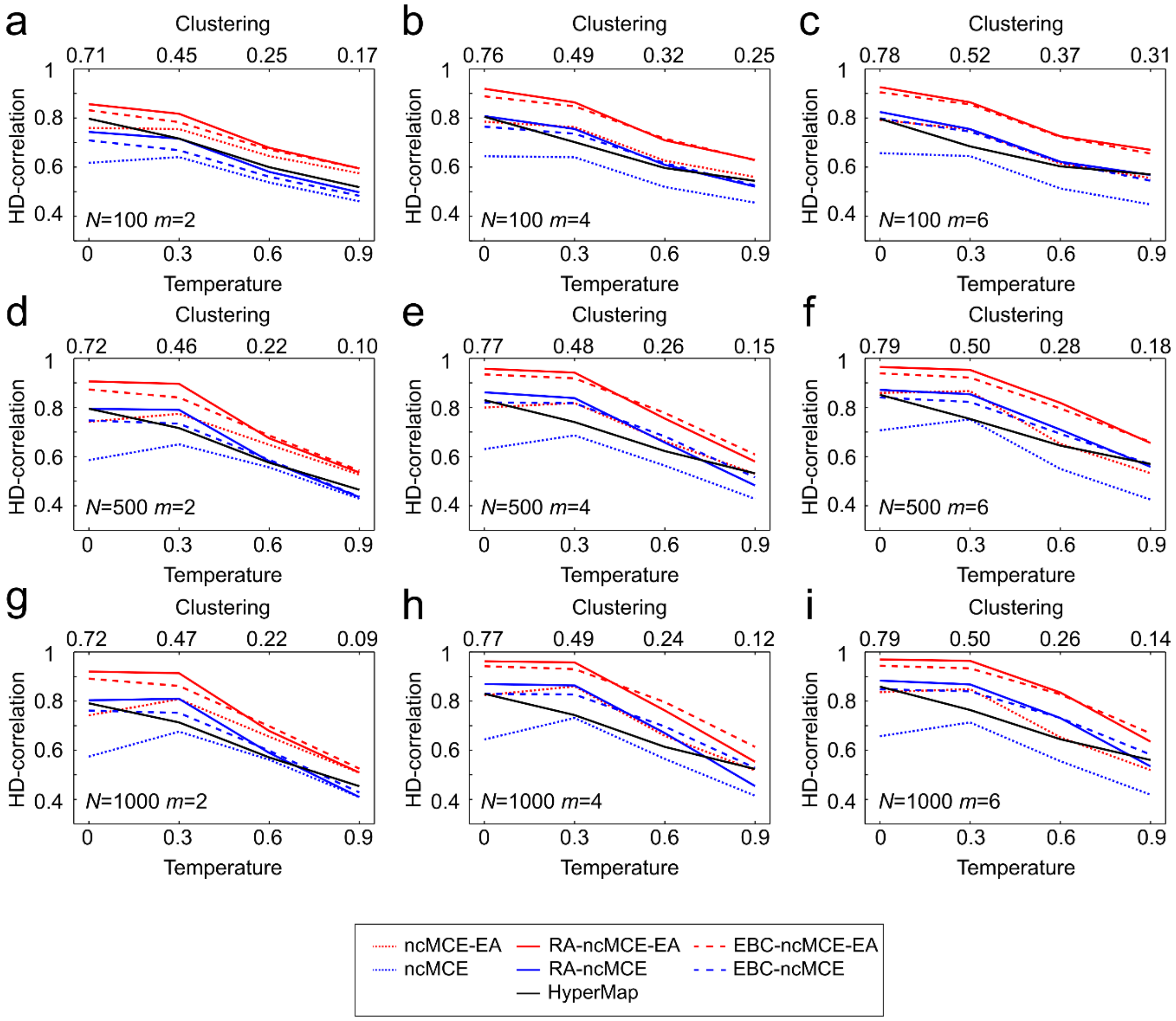


Supplementary Figure 4. HD-correlation on PSO synthetic networks for LE
(a-i) The figure is equivalent to Fig. 3, but all the variants of LE are compared.

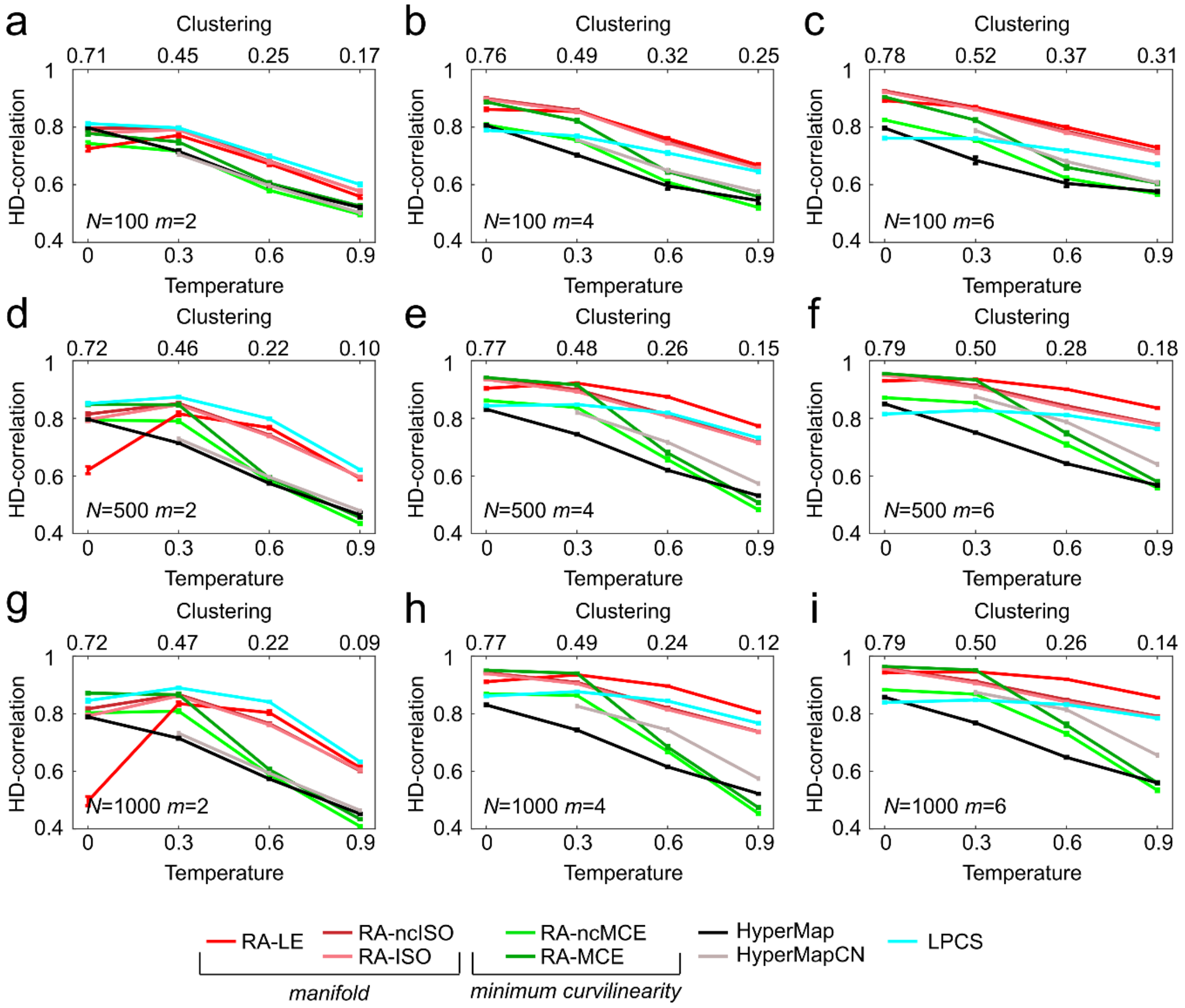


Supplementary Figure 5. HD-correlation on PSO synthetic networks for MCE

(a-i) The figure is equivalent to Fig. 3, but all the variants of MCE are compared.

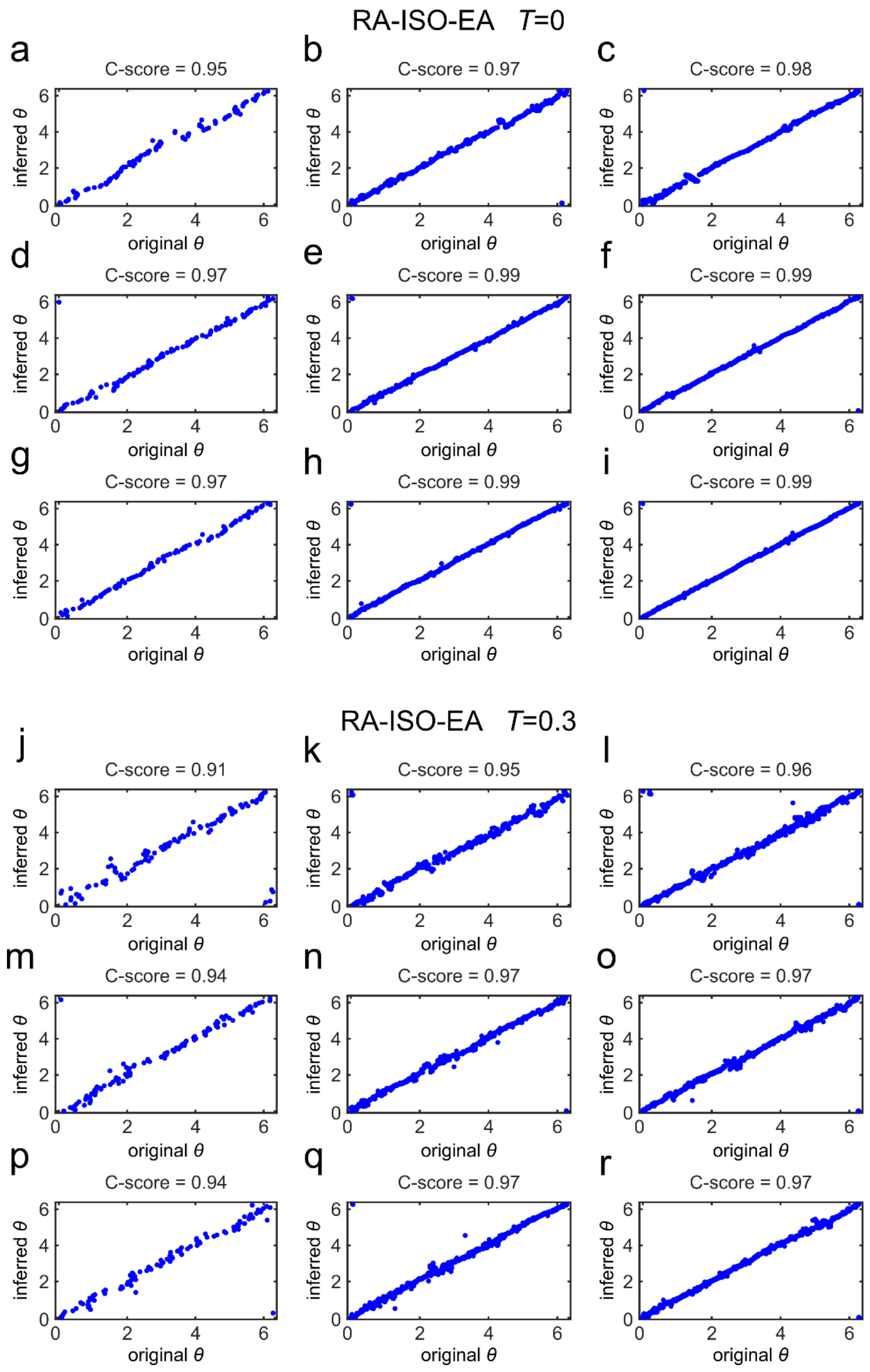


Supplementary Figure 6. HD-correlation on PSO synthetic networks for ncMCE
(a-i) The figure is equivalent to Fig. 3, but all the variants of ncMCE are compared.



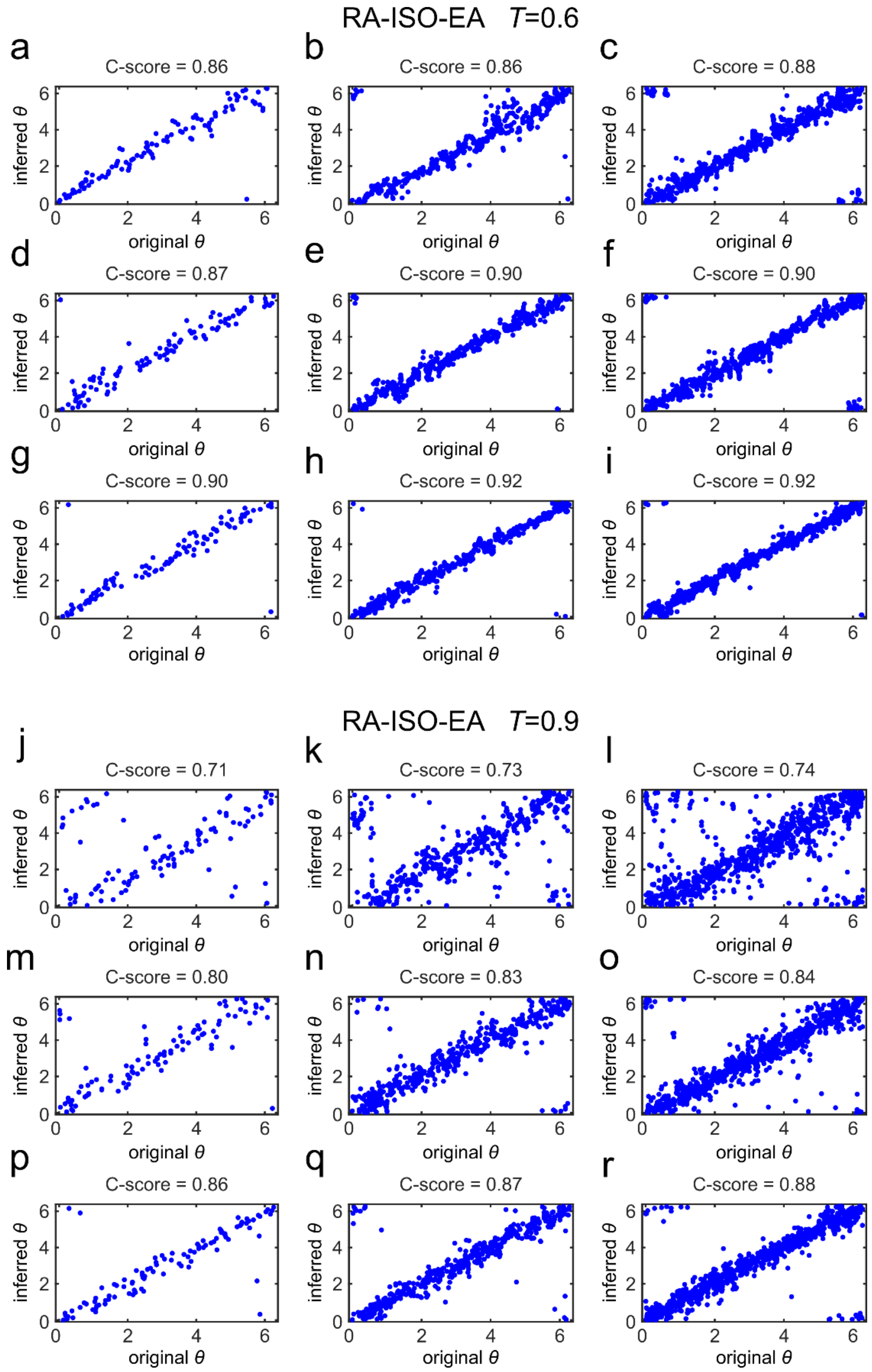
Supplementary Figure 7. HD-correlation on PSO synthetic networks

(a-i) The figure is equivalent to Fig. 3, but the methods without EA are shown. Comparing the two figures, it is evident that the ability of EA to adjust for the local positional noise makes a difference. In fact, when m = 2 and temperatures are low, RA-LE without equidistant adjustment suffers a strong performance reduction in comparison to the case in which EA is applied.

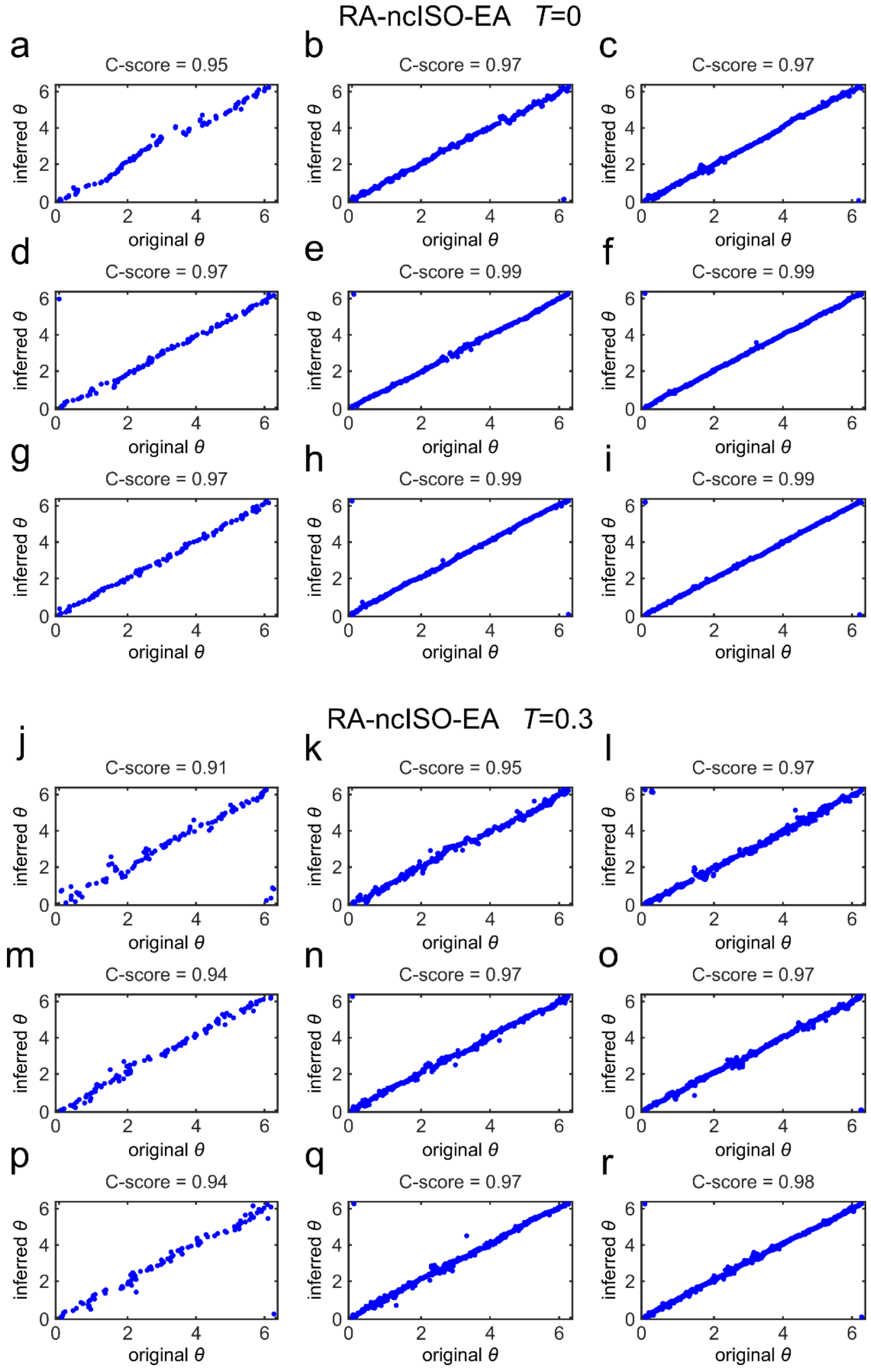


Supplementary Figure 8. Angular coordinates comparison for RA-ISO-EA ($T = 0$, $T = 0.3$)

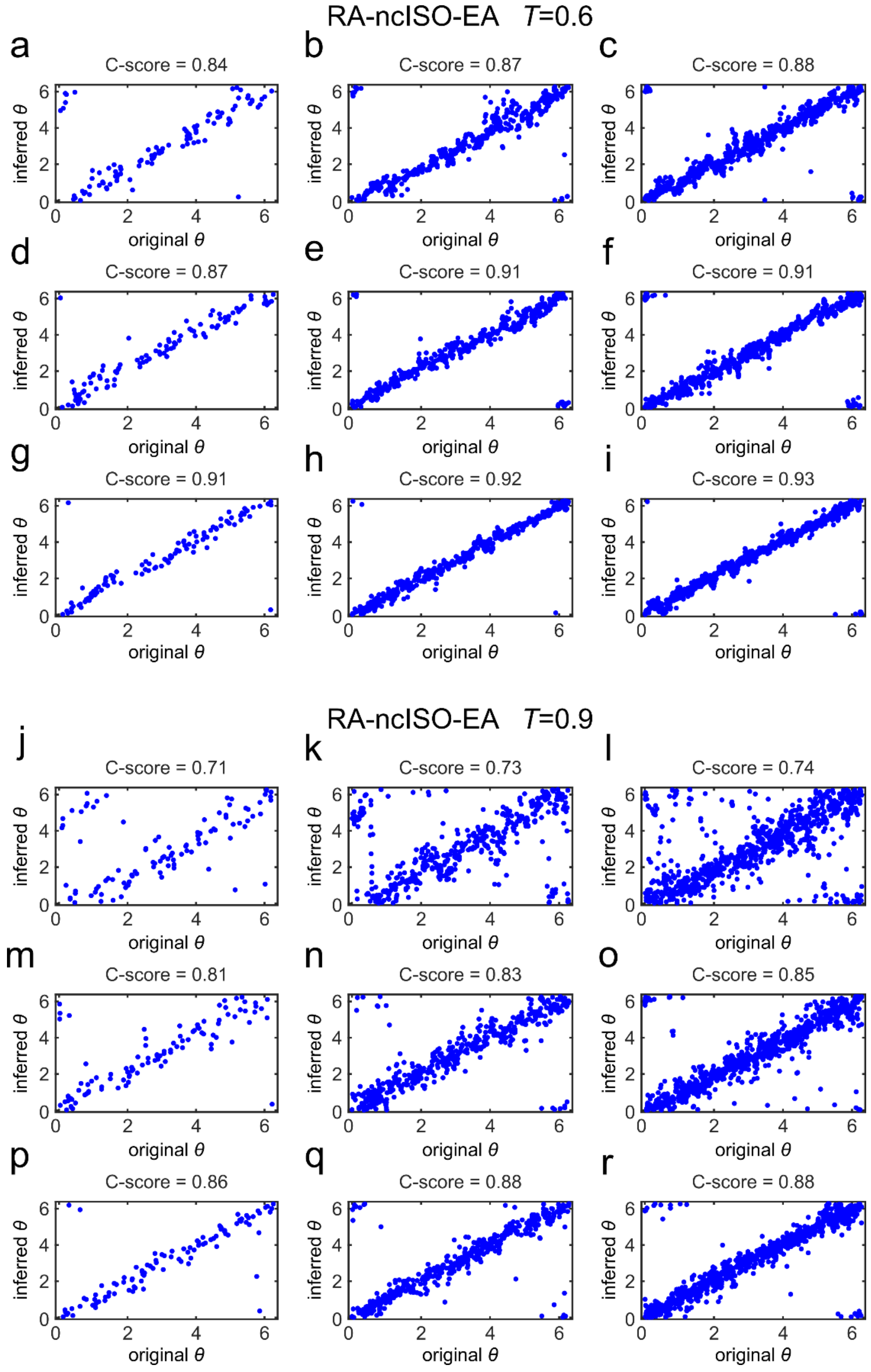
For all the combinations of the PSO parameters N (size) and m (half of average degree), fixing $T = 0$ (a-i) and $T = 0.3$ (j-r), we chose among the synthetic networks embedded with RA-ISO-EA the ones with the best C-score. For these networks we plotted the aligned inferred angular coordinates against the original angular coordinates as described in Fig. 4.



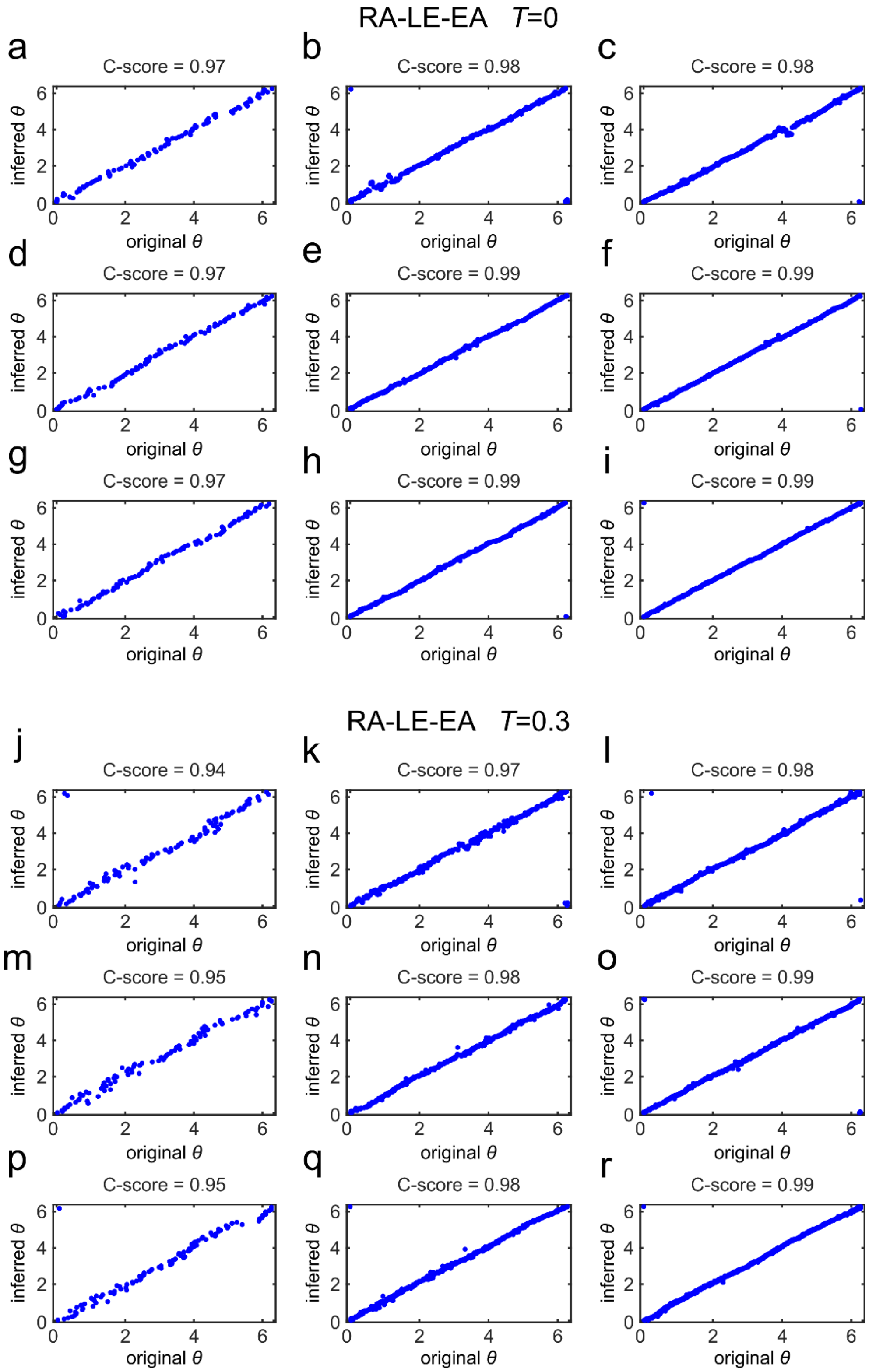
Supplementary Figure 9. Angular coordinates comparison for RA-ISO-EA (*T* = 0.6, *T* = 0.9)
For all the combinations of the PSO parameters *N* (size) and *m* (half of average degree), fixing *T* = 0.6 (a-i) and *T* = 0.9 (j-r), we chose among the synthetic networks embedded with RA-ISO-EA the ones with the best C-score. For these networks we plotted the aligned inferred angular coordinates against the original angular coordinates as described in Fig. 4.



Supplementary Figure 10. Angular coordinates comparison for RA-ncISO-EA (*T* = 0, *T* = 0.3) For all the combinations of the PSO parameters *N* (size) and *m* (half of average degree), fixing *T* = 0 (a-i) and *T* = 0.3 (j-r), we chose among the synthetic networks embedded with RA-ncISO-EA the ones with the best C-score. For these networks we plotted the aligned inferred angular coordinates against the original angular coordinates as described in Fig. 4.

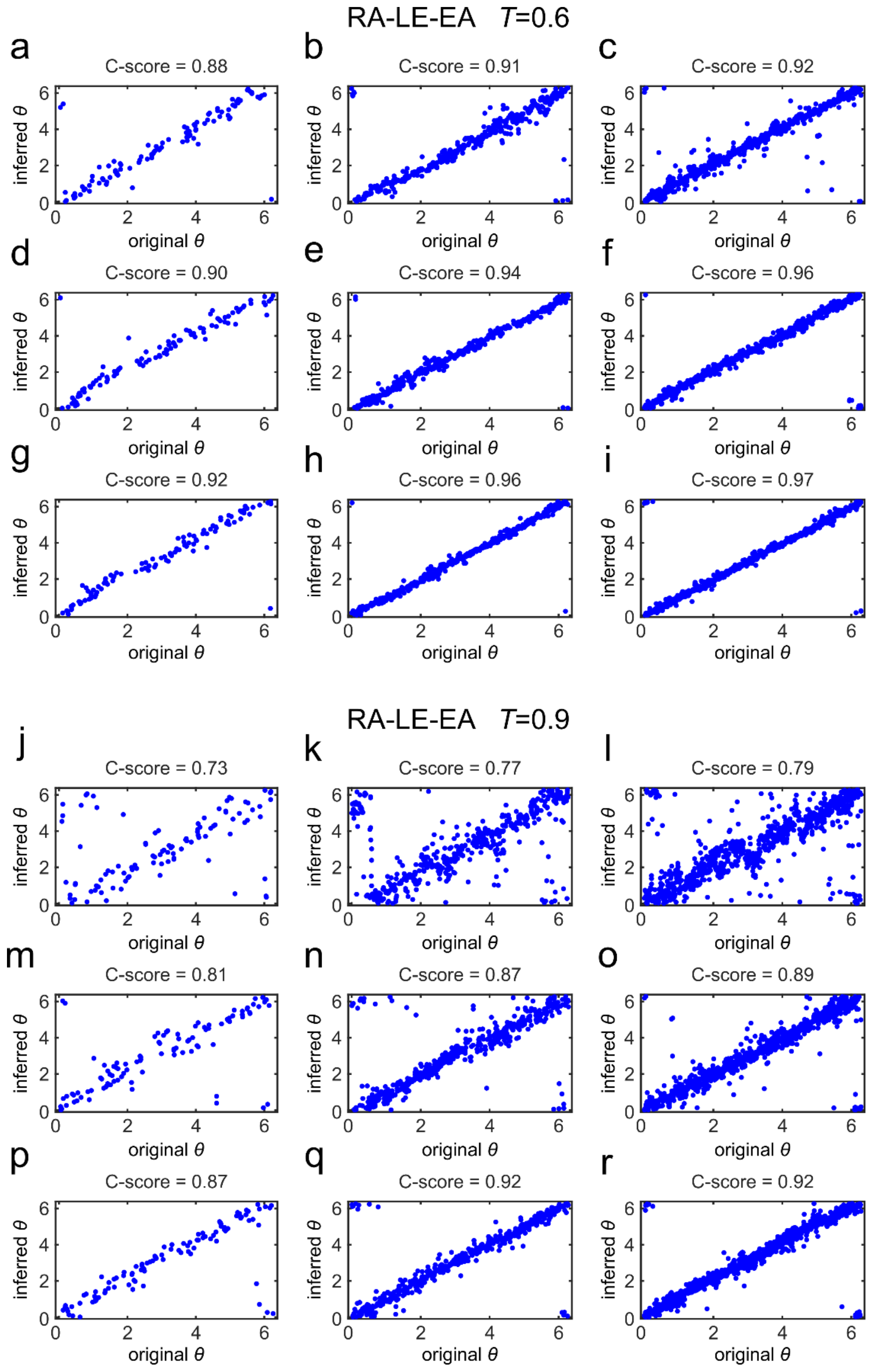


Supplementary Figure 11. Angular coordinates comparison for RA-ncISO-EA (*T* = 0.6, *T* = 0.9)
For all the combinations of the PSO parameters *N* (size) and *m* (half of average degree), fixing *T* = 0.6 (a-i) and *T* = 0.9 (j-r), we chose among the synthetic networks embedded with RA-ncISO-EA the ones with the best C-score. For these networks we plotted the aligned inferred angular coordinates against the original angular coordinates as described in Fig. 4.

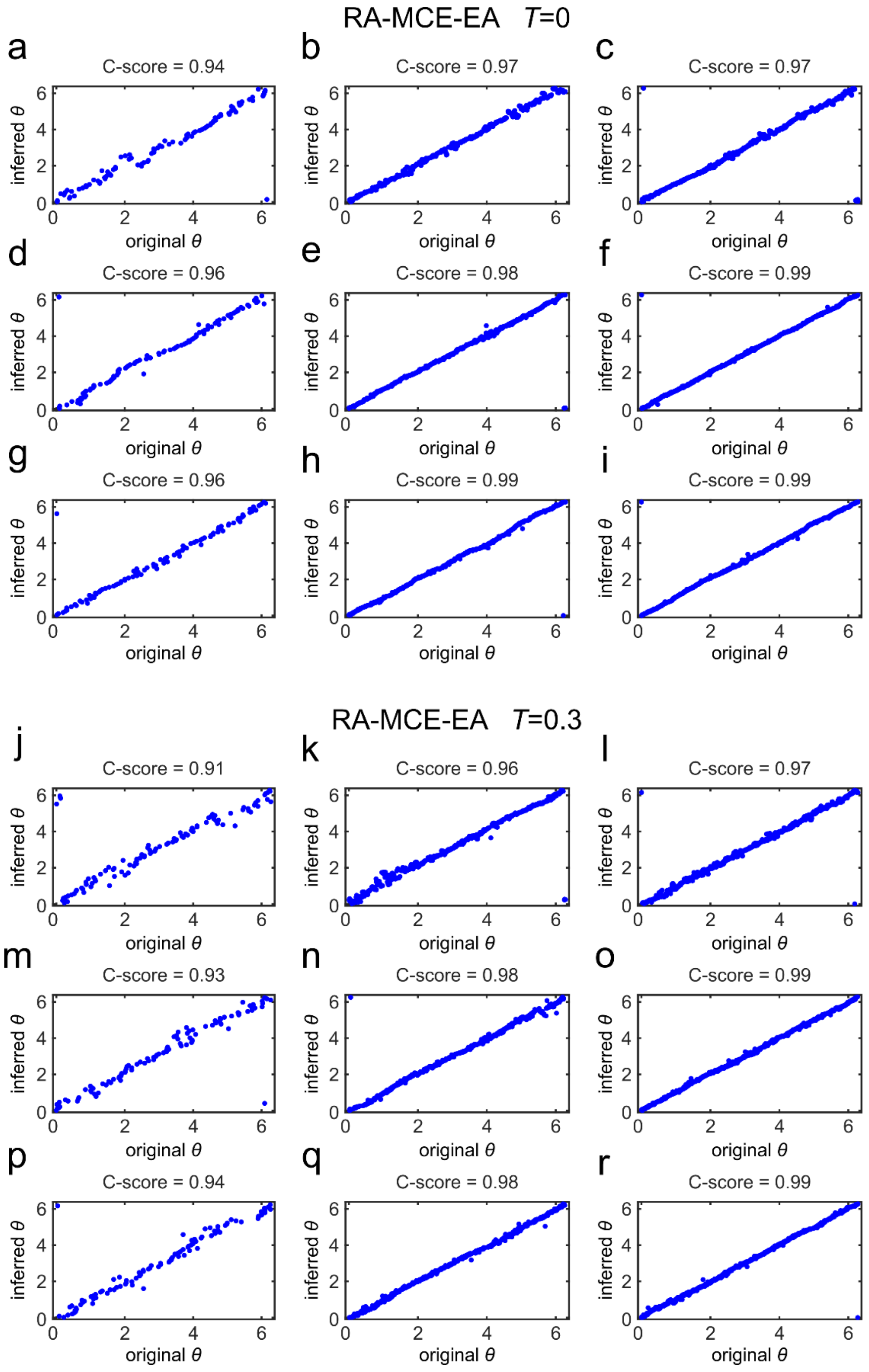


Supplementary Figure 12. Angular coordinates comparison for RA-LE-EA (T = 0, T = 0.3)

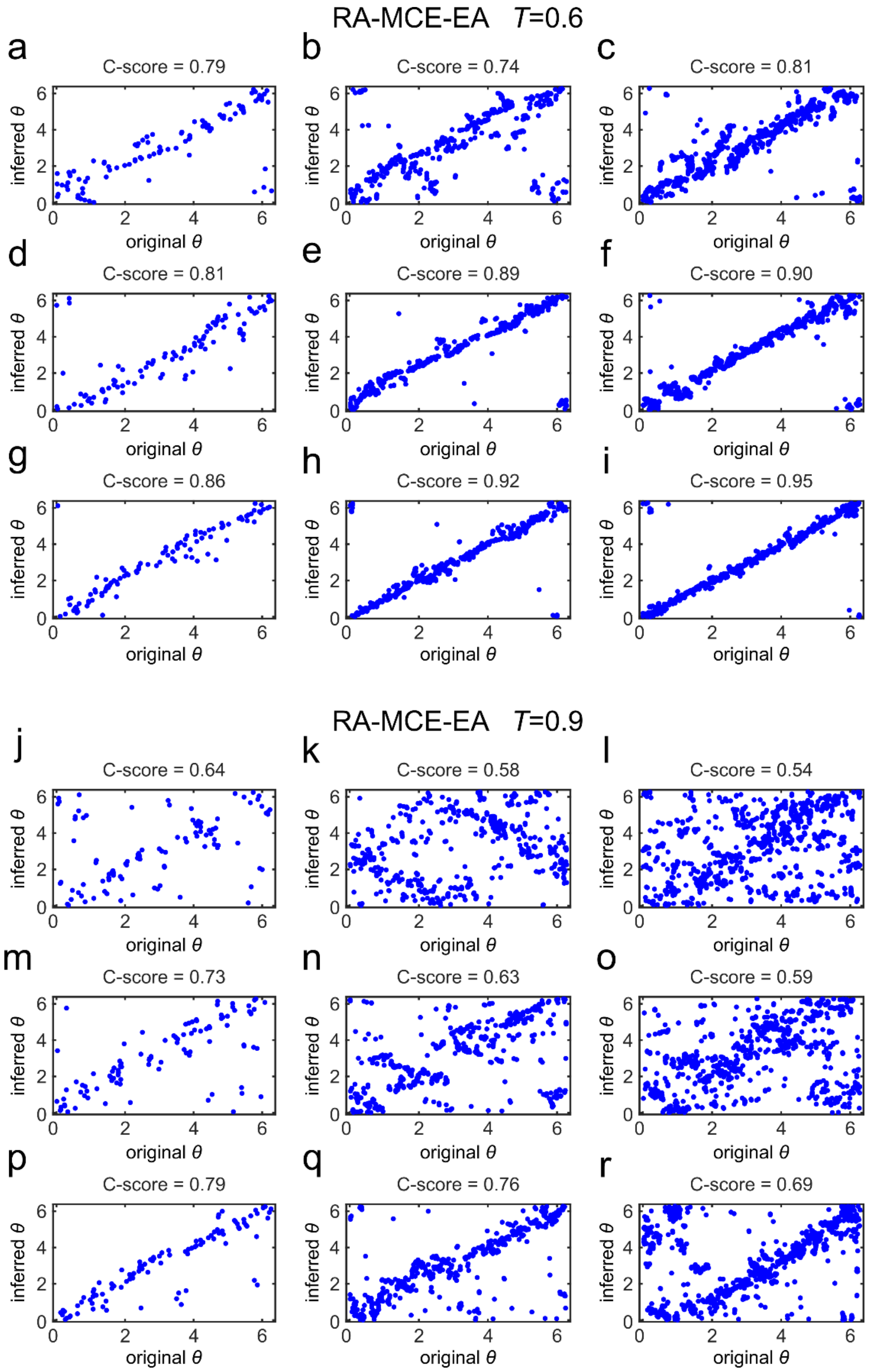
For all the combinations of the PSO parameters N (size) and m (half of average degree), fixing T = 0 (a-i) and T = 0.3 (j-r), we chose among the synthetic networks embedded with RA-LE-EA the ones with the best C-score. For these networks we plotted the aligned inferred angular coordinates against the original angular coordinates as described in Fig. 4.



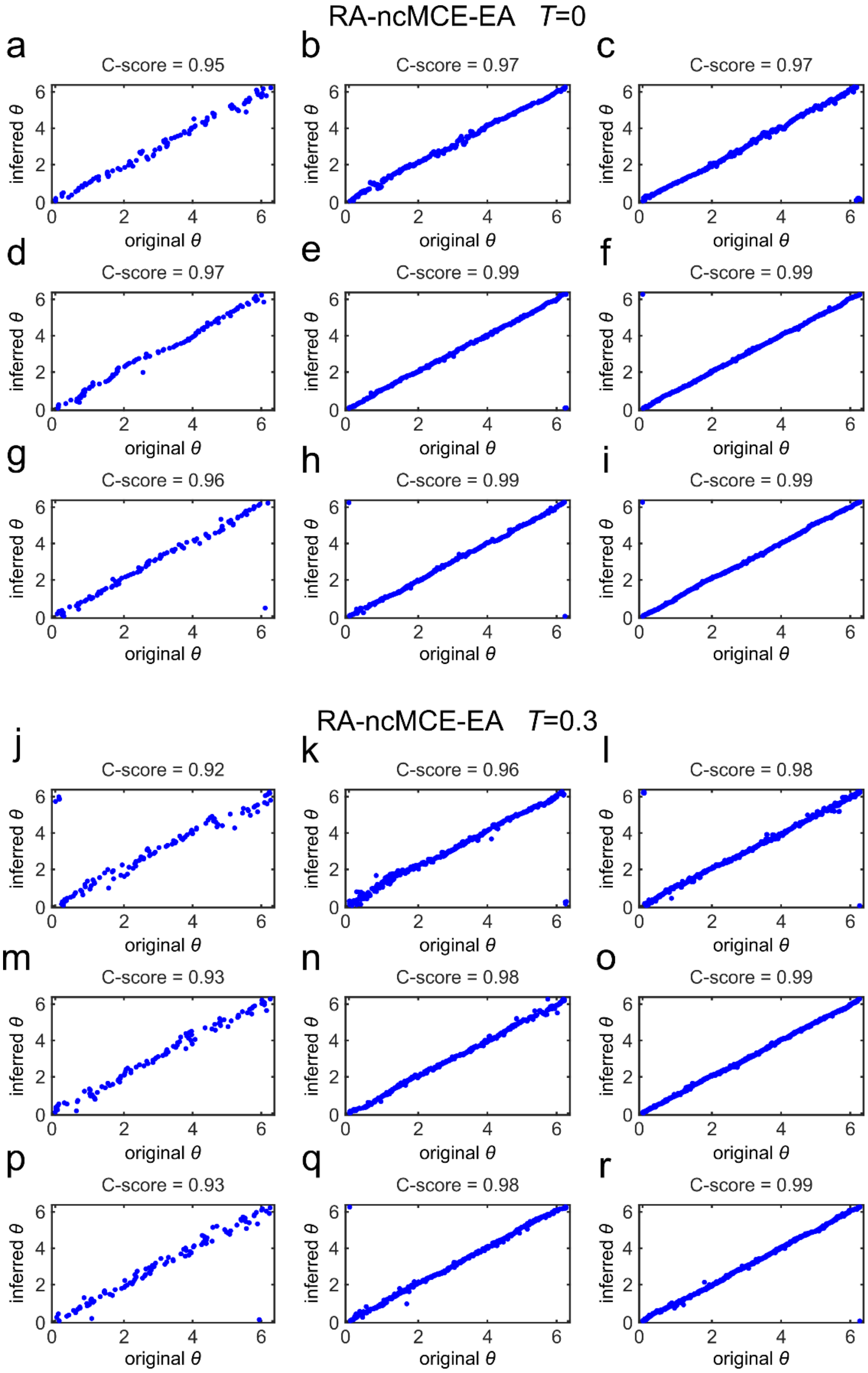
Supplementary Figure 13. Angular coordinates comparison for RA-LE-EA (*T* = 0.6, *T* = 0.9)
For all the combinations of the PSO parameters *N* (size) and *m* (half of average degree), fixing *T* = 0.6 (a-i) and *T* = 0.9 (j-r), we chose among the synthetic networks embedded with RA-LE-EA the ones with the best C-score. For these networks we plotted the aligned inferred angular coordinates against the original angular coordinates as described in Fig. 4.



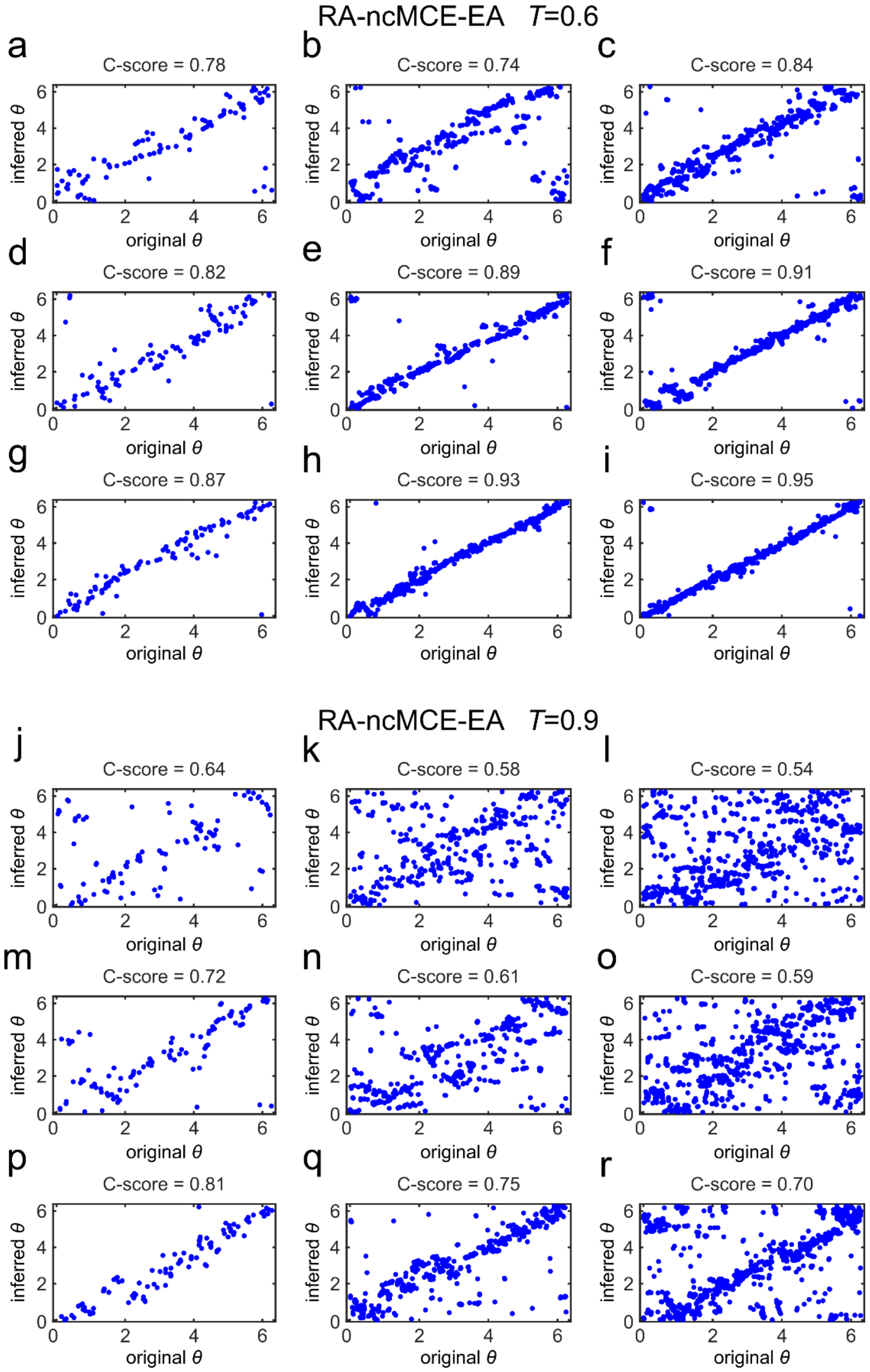
Supplementary Figure 14. Angular coordinates comparison for RA-MCE-EA (*T* = 0, *T* = 0.3)
For all the combinations of the PSO parameters *N* (size) and *m* (half of average degree), fixing *T* = 0 (a-i) and *T* = 0.3 (j-r), we chose among the synthetic networks embedded with RA-MCE-EA the ones with the best C-score. For these networks we plotted the aligned inferred angular coordinates against the original angular coordinates as described in Fig. 4.



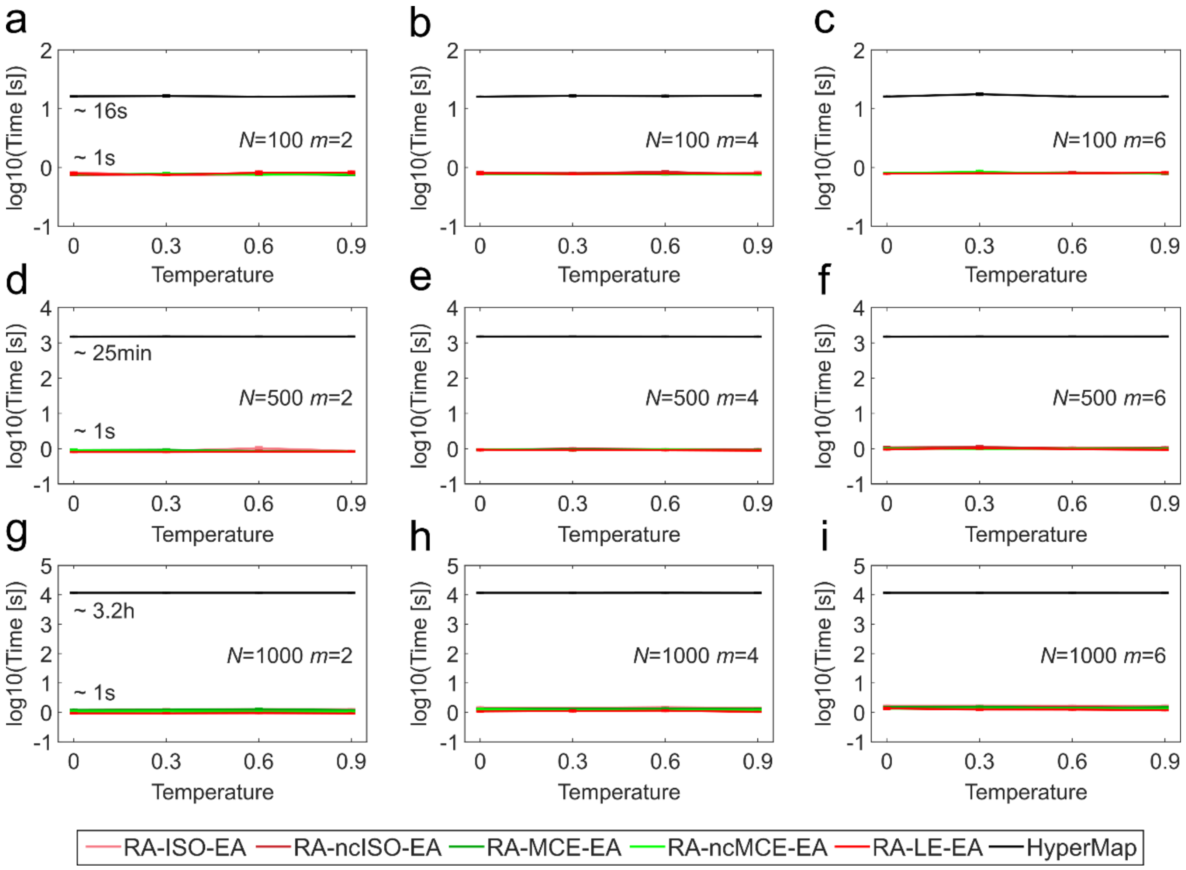
Supplementary Figure 15. Angular coordinates comparison for RA-MCE-EA ($T = 0.6$, $T = 0.9$) For all the combinations of the PSO parameters N (size) and m (half of average degree), fixing $T = 0.6$ (a-i) and $T = 0.9$ (j-r), we chose among the synthetic networks embedded with RA-MCE-EA the ones with the best C-score. For these networks we plotted the aligned inferred angular coordinates against the original angular coordinates as described in Fig. 4.



Supplementary Figure 16. Angular coordinates comparison for RA-ncMCE-EA (T = 0, T = 0.3) For all the combinations of the PSO parameters N (size) and m (half of average degree), fixing T = 0 (a-i) and T = 0.3 (j-r), we chose among the synthetic networks embedded with RA-ncMCE-EA the ones with the best C-score. For these networks we plotted the aligned inferred angular coordinates against the original angular coordinates as described in Fig. 4.

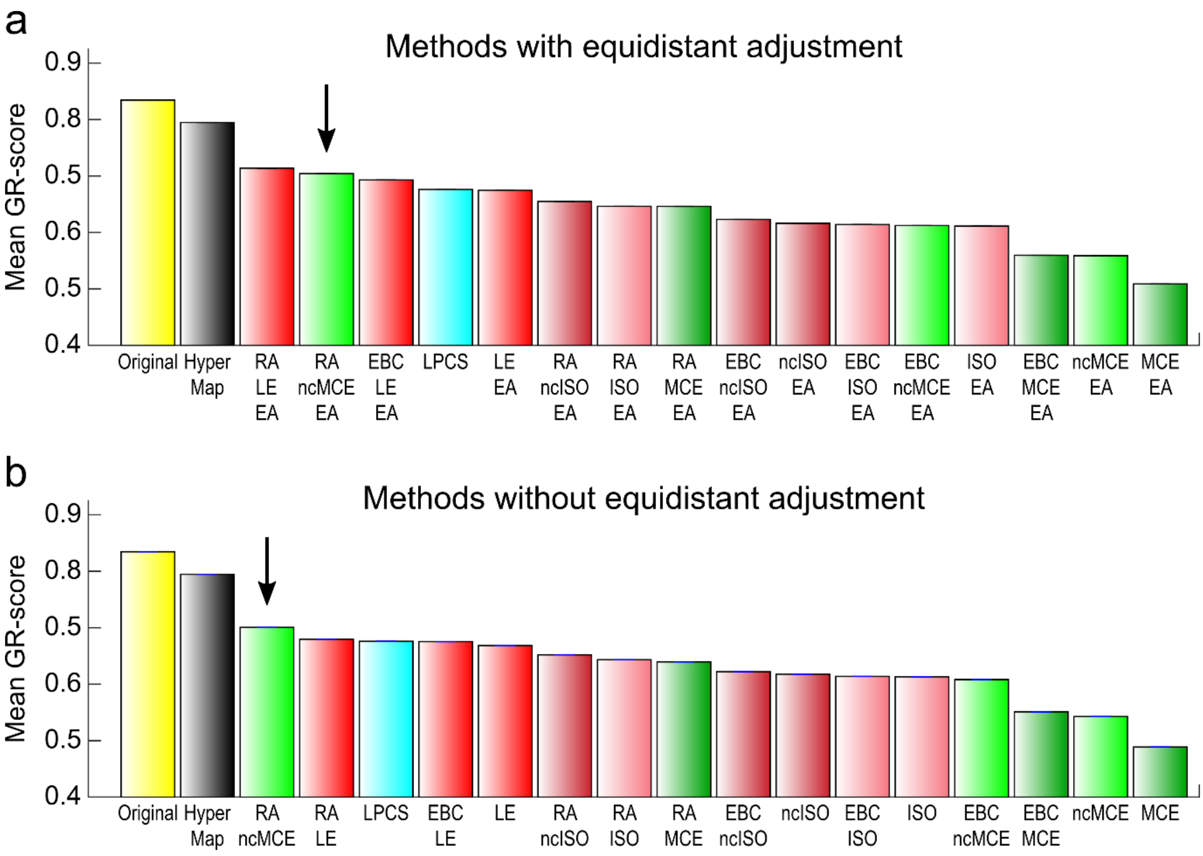


Supplementary Figure 17. Angular coordinates comparison for RA-ncMCE-EA (*T* = 0.6, *T* = 0.9) For all the combinations of the PSO parameters *N* (size) and *m* (half of average degree), fixing *T* = 0.6 (a-i) and *T* = 0.9 (j-r), we chose among the synthetic networks embedded with RA-ncMCE-EA the ones with the best C-score. For these networks we plotted the aligned inferred angular coordinates against the original angular coordinates as described in Fig. 4.



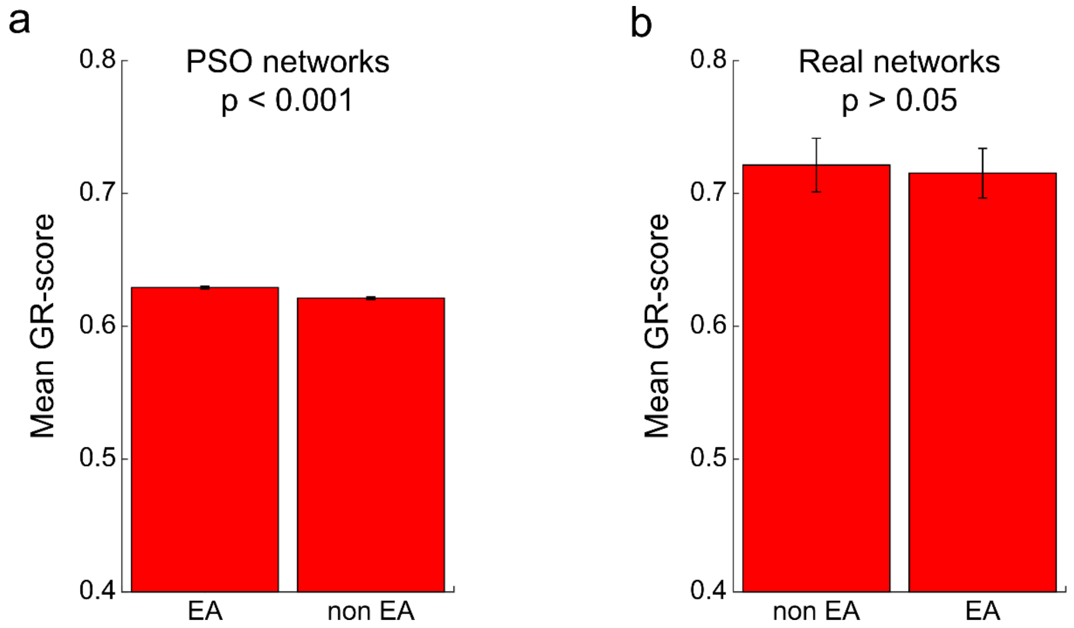
Supplementary Figure 18. Time on PSO synthetic networks

(a-i) For the same PSO networks in Fig. 3, the computational time shows the large efficiency of the coalescent embedding based approaches that generally required around one second to embed networks with 1000 nodes, while HyperMap spent approximately 3 hours for the same task (software and hardware details in the Methods).



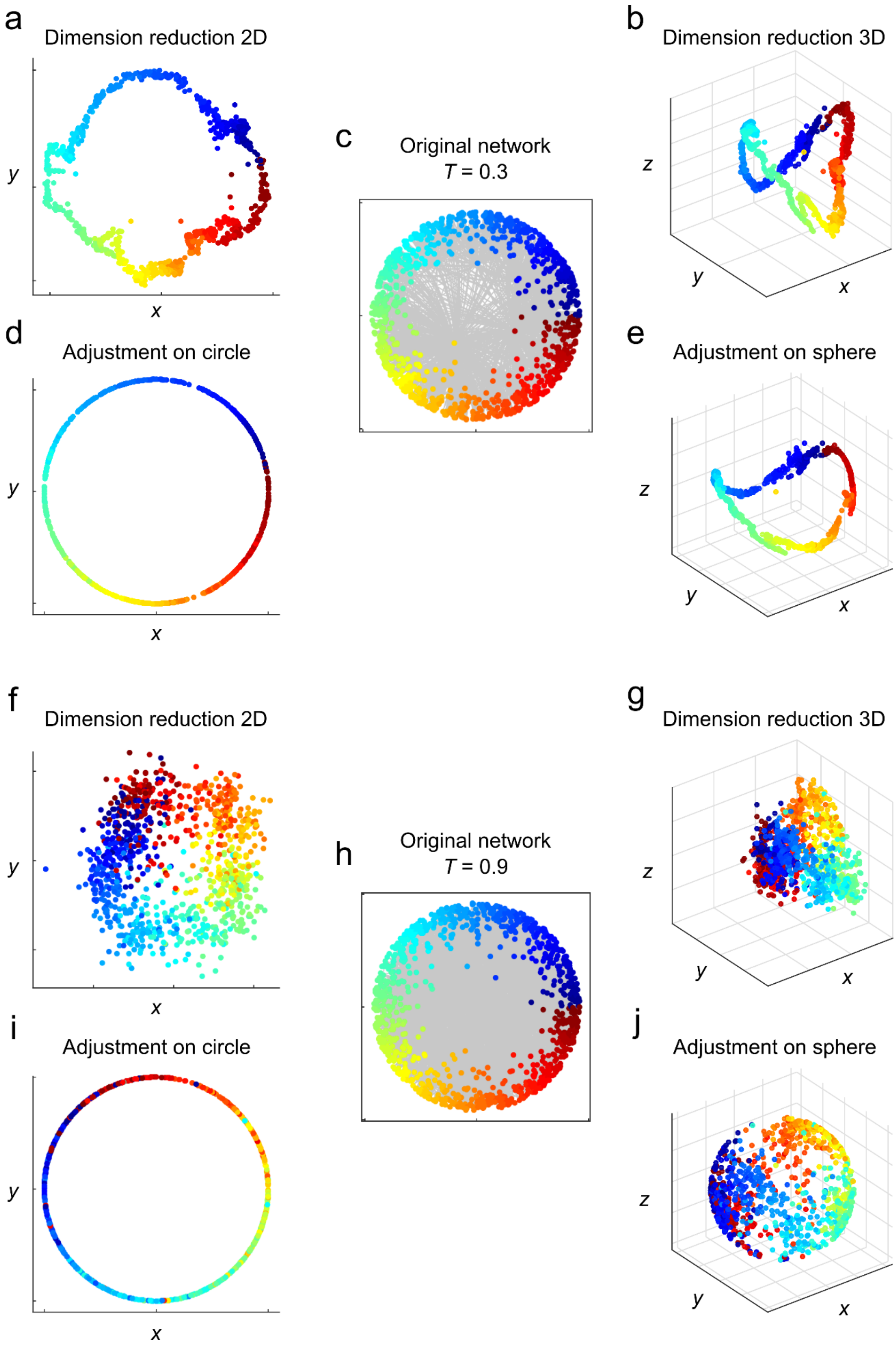
Supplementary Figure 19. Greedy routing on PSO synthetic networks

The same PSO networks considered in Fig. 3 have been mapped using the hyperbolic embedding techniques and the greedy routing in the geometrical space has been evaluated. The barplot reports for each method the mean GR-score over all the PSO parameter combinations. The GR-score is a metric to evaluate the efficiency of the greedy routing, which assumes values between 0, when all the routings are unsuccessful, and 1, when all the packets reach the destination through the shortest path (see Methods for details). Both the EA (a) and non-EA (b) variants are reported, in order to check whether the equidistant adjustment might affect the navigability. The score for HyperMap-CN is not reported since the value for $T = 0$ is missing, because the original code assumes $T > 0$. The GR-score evaluated over the original coordinates of the PSO model is also shown. A black arrow points the coalescent embedding algorithm RA-ncMCE, as in Fig. 6.s

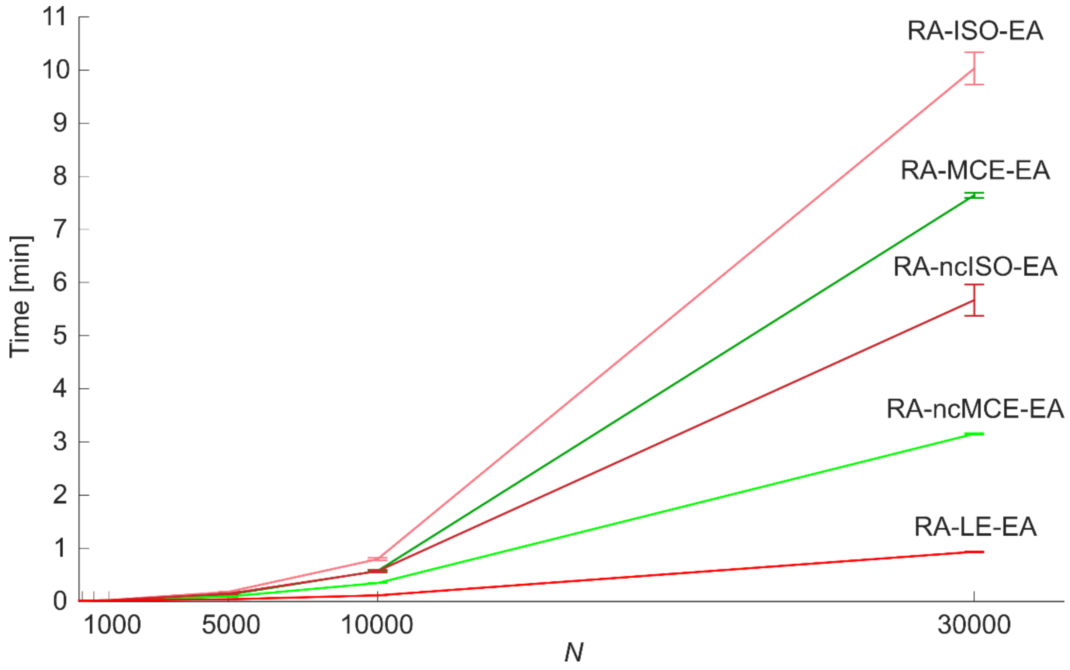


Supplementary Figure 20. Comparison of EA and non-EA for greedy routing

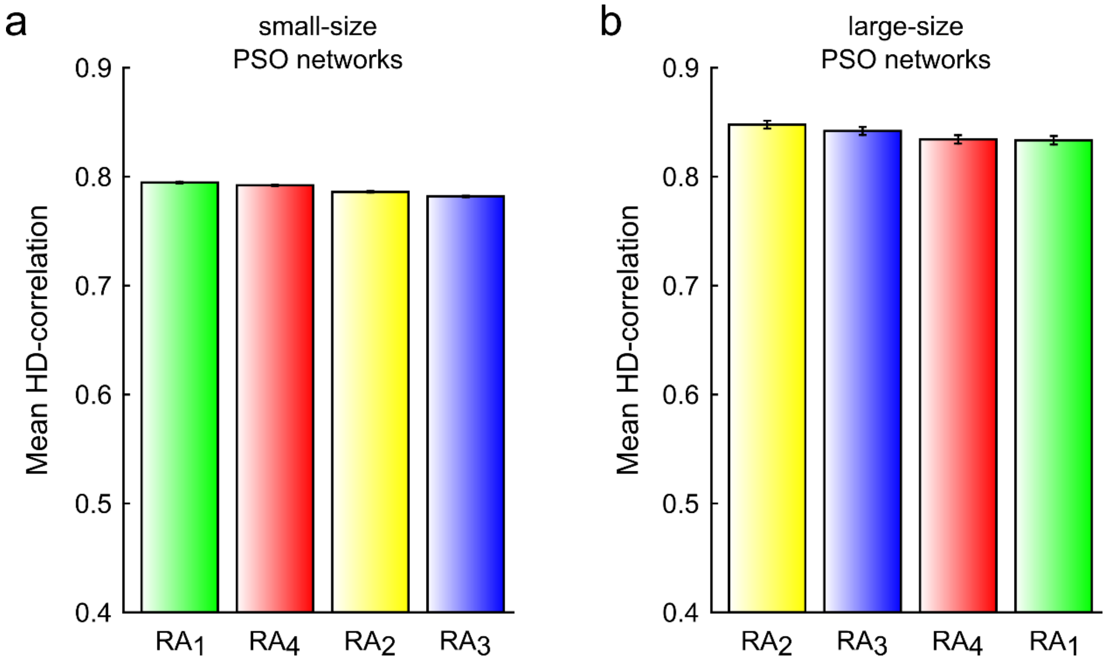
(a) The same PSO networks considered in Fig. 3 have been mapped using the coalescent embedding techniques and the greedy routing in the geometrical space has been evaluated. The barplot reports the GR-score averaged not only over all the PSO parameter combinations but also respectively over all the EA and non-EA coalescent embedding techniques. The corresponding standard error are also shown. The p-value of the permutation test for the mean (10000 iterations) is reported, performed considering the two vectors of GR-scores for the EA and non-EA methods. (b) The 8 real networks whose statistics are reported in Table 1 have been mapped using the coalescent embedding techniques and the greedy routing in the geometrical space has been evaluated. The barplot reports the GR-score averaged not only over the networks but also respectively over all the EA and non-EA coalescent embedding techniques. The corresponding standard error are also shown. The p-value of the permutation test for the mean (10000 iterations) is reported, performed considering the two vectors of GR-scores for the EA and non-EA methods.



Supplementary Figure 21. Comparison of 2D and 3D embedding for increasing temperature
The figure shows how the similarities of the original PSO network (N = 1000, m = 6, γ = 2.5) are recovered either embedding in 2D (a, f) and arranging the angular coordinates over the circumference of a circle (d, i) or embedding in 3D (b, g) and adjusting the angular coordinates over a sphere (e, j). The figure reports the plots for the temperatures T = 0.3 (a-e) and T = 0.9 (f-j) in order to integrate the Fig. 8, where T = 0 and T = 0.6 are shown.

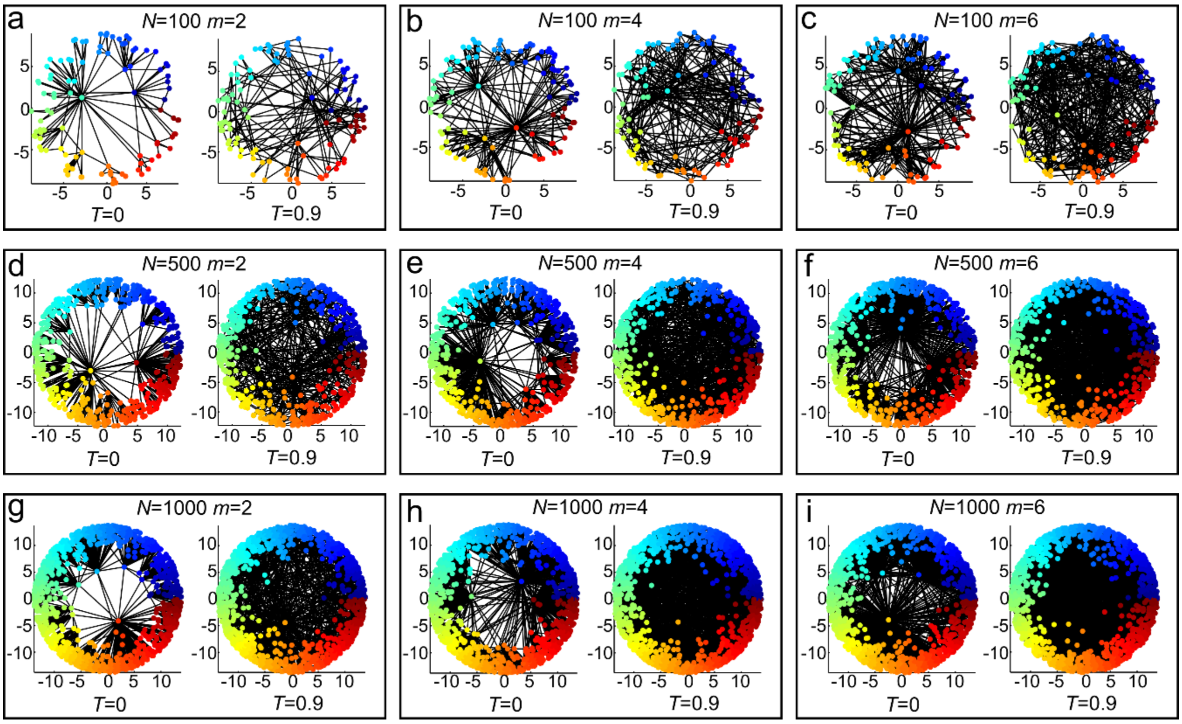


Supplementary Figure 22. Time estimation for coalescent embedding methods on PSO networks For the PSO parameters $\gamma = 2.5$, $T = 0.3$ (since Supplementary Fig. 4 showed that the time is T invariant) and $m = [2, 4, 6]$, we generated 10 networks of size $N = [100, 500, 1000, 5000, 10000]$ and 5 networks of size $N = 30000$. The plot reports the average embedding time for increasing network size and for each of the coalescent embedding techniques with RA pre-weighting and equidistant adjustment. The corresponding standard error are also shown.

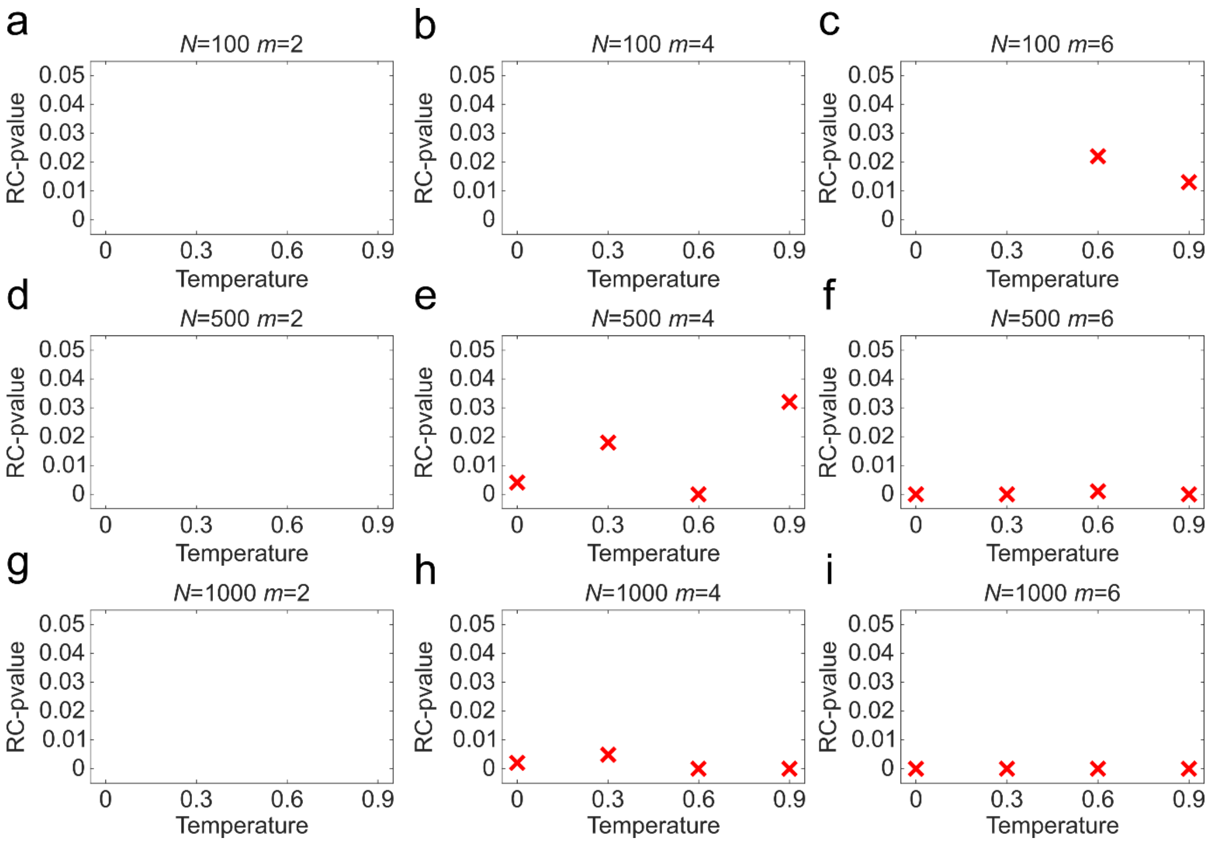


Supplementary Figure 23. Comparison of Repulsion-Attraction (RA) pre-weighting rules

Since there was not a unique possible mathematical formulation of the RA formula, we tested four variants differing in the way in which the degrees of the connected nodes are combined. For each combination of the PSO parameters m = [2, 4, 6] and T = [0, 0.3, 0.6, 0.9], fixing γ = 2.5, we generated 100 networks of size N = [100, 500, 1000], 10 networks of size N = 10000 and 5 networks of size N = 30000. The networks have been embedded using the coalescent embedding techniques applying the different RA pre-weighting variants and the HD-correlation has been evaluated. The figure reports for each RA variant the HD-correlation averaged over the PSO parameter combinations and over the different coalescent embedding techniques, for networks of size N = [100, 500, 1000] (a) and N = [10000, 30000] (b). The corresponding standard error are also shown.



Supplementary Figure 24. Examples of synthetic networks generated by the PSO model
(a-i) For each combination of the parameters *N* (size) and *m* (half of average degree), examples of synthetic networks with temperature $T = 0$ and $T = 0.9$ are shown, in order to illustrate the increase of randomness and the loss of the tree like structure for high temperatures.



Supplementary Figure 25. Rich-club statistical test on PSO synthetic networks

For each combination of the PSO parameters *N*, *m* and *T*, the statistical test for rich-clubness [110] has been performed on 10 networks and the p-values have been adjusted for multiple hypothesis testing by the Bonferroni correction. For each parameter combination the average of the adjusted p-values is reported, highlighting the range below the significance level of 0.05.

Method	Karate	Opsahl 8	Opsahl 9	Opsahl 10	Opsahl 11	Polbooks	Football	Polblogs	Mean	% Impr.
	N=34	N=43	N=44	N=77	N=77	N=105	N=115	N=1222		
	E=78	E=193	E=348	E=518	E=1088	E=441	E=613	E=16714		
	m=2.29	m=4.49	m=7.91	m=6.73	m=14.13	m=4.20	m=5.33	m=13.68		
	T=0.43	T=0.43	T=0.32	T=0.35	T=0.28	T=0.51	T=0.60	T=0.68		
	$\boldsymbol{\gamma}$=2.12	γ=8.20	γ=5.92	γ=5.06	γ=4.87	γ=2.62	γ=9.09	γ=2.38		
	N_c=2	N_c=7	N_c=7	N_c=4	N_c=4	N_c=3	N_c=12	N_c=2		
EBC-ncISO-EA	1.00	0.57	0.47	1.00	0.93	0.59	0.90	0.68	0.77	+13.2
RA-MCE-EA	0.83	0.51	0.47	1.00	0.96	0.57	0.82	0.67	0.73	+7.4
RA-ncMCE-EA	0.73	0.55	0.47	1.00	1.00	0.57	0.83	0.67	0.73	+7.4
EBC-MCE-EA	0.83	0.47	0.41	1.00	0.96	0.57	0.90	0.62	0.72	+5.9
EBC-ncMCE-EA	0.88	0.46	0.41	1.00	0.96	0.57	0.85	0.62	0.72	+5.9
EBC-ISO-EA	0.83	0.42	0.47	1.00	0.89	0.59	0.88	0.66	0.72	+5.9
LPCS	0.83	0.49	0.41	1.00	0.96	0.55	0.87	0.67	0.72	+5.9
ncMCE-EA	0.73	0.47	0.47	1.00	0.96	0.57	0.89	0.62	0.71	+4.4
RA-LE-EA	0.67	0.48	0.53	1.00	0.92	0.56	0.82	0.70	0.71	+4.4
EBC-ISO	0.83	0.47	0.47	1.00	0.76	0.54	0.89	0.68	0.71	+4.4
RA-MCE	0.83	0.52	0.36	0.93	0.92	0.56	0.85	0.66	0.71	+4.4
RA-ncISO-EA	0.67	0.54	0.42	1.00	0.92	0.56	0.86	0.67	0.70	+2.9
ncISO-EA	0.73	0.50	0.41	1.00	0.88	0.54	0.87	0.66	0.70	+2.9
EBC-LE-EA	0.85	0.42	0.41	0.96	0.92	0.56	0.85	0.62	0.70	+2.9
RA-ncISO	0.68	0.46	0.41	1.00	0.87	0.54	0.89	0.72	0.70	+2.9
EBC-MCE	0.83	0.42	0.51	1.00	0.92	0.59	0.86	0.40	0.69	+1.5
EBC-ncISO	0.68	0.49	0.36	1.00	0.85	0.57	0.85	0.71	0.69	+1.5
MCE-EA	0.64	0.47	0.47	0.96	0.92	0.55	0.86	0.62	0.69	+1.5
RA-ISO	0.68	0.43	0.39	1.00	0.87	0.54	0.87	0.70	0.69	+1.5
unweighted	0.46	0.55	0.41	1.00	0.96	0.50	0.93	0.64	0.68	0.0
LE	0.68	0.53	0.37	1.00	0.79	0.53	0.85	0.73	0.68	0.0
LE-EA	0.63	0.55	0.41	1.00	0.78	0.55	0.82	0.67	0.68	0.0
RA-ISO-EA	0.57	0.43	0.44	1.00	0.88	0.54	0.86	0.67	0.67	-1.5
ISO	0.59	0.49	0.45	0.96	0.85	0.56	0.80	0.68	0.67	-1.5
RA-ncMCE	0.67	0.45	0.41	0.93	0.89	0.49	0.84	0.64	0.67	-1.5
RA-LE	0.55	0.40	0.44	0.96	0.83	0.56	0.85	0.71	0.66	-2.9
ncISO	0.68	0.55	0.41	0.96	0.79	0.56	0.68	0.67	0.66	-2.9
ISO-EA	0.34	0.50	0.41	0.96	0.93	0.56	0.82	0.67	0.65	-4.4
HyperMap	0.56	0.60	0.28	0.92	0.85	0.50	0.83	0.69	0.65	-4.4
HyperMapCN	0.55	0.47	0.41	0.93	0.79	0.54	0.79	0.70	0.65	-4.4
EBC-LE	0.68	0.36	0.36	1.00	0.79	0.54	0.82	0.60	0.64	-5.9
EBC-ncMCE	0.57	0.37	0.49	1.00	0.89	0.56	0.82	0.22	0.62	-8.8
MCE	0.73	0.28	0.32	0.50	0.21	0.54	0.77	0.42	0.47	-30.9
ncMCE	0.61	0.24	0.25	0.49	0.18	0.49	0.79	0.08	0.39	-42.6

Supplementary Table 1. Community detection on real networks with Louvain algorithm (HSP-kernel). The table is equivalent to Table 1, but also the non-EA methods are shown.

Method	Karate	Opsahl 8	Opsahl 9	Opsahl 10	Opsahl 11	Polbooks	Football	Polblogs	Mean	% Impr.
	N=34	N=43	N=44	N=77	N=77	N=105	N=115	N=1222		
	E=78	E=193	E=348	E=518	E=1088	E=441	E=613	E=16714		
	m=2.29	m=4.49	m=7.91	m=6.73	m=14.13	m=4.20	m=5.33	m=13.68		
	T=0.43	T=0.43	T=0.32	T=0.35	T=0.28	T=0.51	T=0.60	T=0.68		
	$\boldsymbol{\gamma}$=2.12	γ=8.20	γ=5.92	γ=5.06	γ=4.87	γ=2.62	γ=9.09	γ=2.38		
	N_c=2	N_c=7	N_c=7	N_c=4	N_c=4	N_c=3	N_c=12	N_c=2		
EBC-ncISO-EA	0.68	0.75	0.47	1.00	1.00	0.54	0.92	0.53	0.74	+4.2
ncMCE-EA	0.68	0.74	0.47	1.00	0.93	0.50	0.92	0.52	0.72	+1.4
unweighted	0.55	0.69	0.47	1.00	1.00	0.52	0.92	0.52	0.71	0.0
EBC-MCE-EA	0.68	0.55	0.53	1.00	0.96	0.52	0.93	0.52	0.71	0.0
EBC-ncMCE-EA	0.58	0.55	0.53	1.00	1.00	0.52	0.93	0.52	0.70	-1.4
ISO-EA	0.68	0.53	0.47	1.00	0.96	0.52	0.92	0.53	0.70	-1.4
LE-EA	0.68	0.54	0.47	1.00	0.96	0.53	0.92	0.51	0.70	-1.4
EBC-LE-EA	0.68	0.55	0.47	0.95	0.96	0.52	0.93	0.53	0.70	-1.4
EBC-ncISO	0.68	0.52	0.47	1.00	0.93	0.51	0.92	0.54	0.70	-1.4
RA-ISO	0.58	0.55	0.47	1.00	1.00	0.52	0.93	0.51	0.70	-1.4
ncISO-EA	0.68	0.53	0.47	1.00	0.96	0.47	0.92	0.53	0.69	-2.8
EBC-ISO-EA	0.55	0.55	0.47	1.00	1.00	0.52	0.92	0.53	0.69	-2.8
MCE-EA	0.68	0.54	0.47	0.95	0.93	0.51	0.92	0.52	0.69	-2.8
RA-ncISO-EA	0.55	0.55	0.47	1.00	1.00	0.52	0.92	0.52	0.69	-2.8
RA-ISO-EA	0.58	0.55	0.47	1.00	0.96	0.52	0.92	0.52	0.69	-2.8
RA-ncMCE-EA	0.47	0.55	0.53	1.00	1.00	0.52	0.92	0.50	0.69	-2.8
EBC-ISO	0.57	0.55	0.47	1.00	1.00	0.45	0.92	0.53	0.69	-2.8
LPCS	0.55	0.55	0.53	1.00	0.96	0.52	0.93	0.51	0.69	-2.8
LE	0.63	0.55	0.36	1.00	1.00	0.48	0.92	0.54	0.68	-4.2
RA-LE-EA	0.55	0.55	0.47	1.00	0.93	0.52	0.92	0.52	0.68	-4.2
EBC-LE	0.68	0.55	0.36	1.00	0.94	0.47	0.92	0.53	0.68	-4.2
EBC-ncMCE	0.54	0.55	0.49	1.00	0.93	0.48	0.93	0.52	0.68	-4.2
RA-ncISO	0.54	0.61	0.36	1.00	0.96	0.52	0.92	0.52	0.68	-4.2
RA-MCE-EA	0.47	0.55	0.53	1.00	0.92	0.52	0.92	0.51	0.68	-4.2
ncISO	0.54	0.53	0.36	1.00	1.00	0.52	0.92	0.52	0.67	-5.6
ISO	0.58	0.53	0.36	0.96	0.96	0.52	0.93	0.54	0.67	-5.6
RA-MCE	0.47	0.55	0.53	0.93	0.92	0.52	0.92	0.51	0.67	-5.6
EBC-MCE	0.58	0.55	0.22	1.00	0.96	0.52	0.93	0.52	0.66	-7.0
HyperMapCN	0.52	0.55	0.41	1.00	0.86	0.57	0.89	0.46	0.66	-7.0
HyperMap	0.52	0.60	0.32	1.00	0.92	0.49	0.90	0.46	0.65	-8.5
RA-LE	0.43	0.55	0.36	1.00	0.96	0.45	0.92	0.53	0.65	-8.5
RA-ncMCE	0.47	0.61	0.36	0.93	0.96	0.42	0.92	0.44	0.64	-9.9
MCE	0.57	0.53	0.55	0.93	0.50	0.59	0.92	0.38	0.62	-12.7
ncMCE	0.54	0.45	0.59	0.98	0.50	0.41	0.92	0.38	0.60	-15.5

Supplementary Table 2. Community detection on real networks with Infomap algorithm. The table is equivalent to Table 2, but also the non-EA methods are shown.

Method	Karate	Opsahl 8	Opsahl 9	Opsahl 10	Opsahl 11	Polbooks	Football	Polblogs	Mean	% Impr.
	N=34	N=43	N=44	N=77	N=77	N=105	N=115	N=1222		
	E=78	E=193	E=348	E=518	E=1088	E=441	E=613	E=16714		
	m=2.29	m=4.49	m=7.91	m=6.73	m=14.13	m=4.20	m=5.33	m=13.68		
	T=0.43	T=0.43	T=0.32	T=0.35	T=0.28	T=0.51	T=0.60	T=0.68		
	γ=2.12	γ=8.20	γ=5.92	γ=5.06	γ=4.87	γ=2.62	γ=9.09	γ=2.38		
	N_c=2	N_c=7	N_c=7	N_c=4	N_c=4	N_c=3	N_c=12	N_c=2		
ISO	0.63	0.50	0.25	0.94	0.83	0.56	0.91	0.69	0.67	+9.8
EBC-ncISO-EA	0.72	0.53	0.42	0.96	0.43	0.52	0.91	0.68	0.65	+6.6
ncISO-EA	0.75	0.38	0.50	0.95	0.44	0.52	0.89	0.69	0.64	+4.9
EBC-ncISO	0.52	0.58	0.32	1.00	0.57	0.51	0.91	0.68	0.64	+4.9
RA-ISO	0.58	0.46	0.42	1.00	0.54	0.50	0.90	0.68	0.63	+3.3
RA-ncMCE-EA	0.68	0.51	0.53	0.97	0.29	0.54	0.90	0.64	0.63	+3.3
EBC-ISO	0.50	0.40	0.42	0.99	0.66	0.51	0.90	0.66	0.63	+3.3
EBC-ncMCE	0.42	0.47	0.40	0.97	0.70	0.49	0.91	0.69	0.63	+3.3
RA-MCE	0.62	0.41	0.38	0.99	0.71	0.51	0.87	0.55	0.63	+3.3
RA-LE	0.43	0.40	0.28	1.00	0.83	0.49	0.91	0.68	0.63	+3.3
RA-ncISO	0.44	0.51	0.24	1.00	0.73	0.52	0.89	0.69	0.63	+3.3
LE	0.63	0.49	0.27	1.00	0.48	0.51	0.91	0.70	0.62	+1.6
EBC-LE-EA	0.76	0.46	0.34	0.92	0.38	0.54	0.91	0.68	0.62	+1.6
LPCS	0.49	0.48	0.29	1.00	0.60	0.53	0.89	0.68	0.62	+1.6
unweighted	0.73	0.44	0.27	0.99	0.34	0.56	0.88	0.70	0.61	0.0
HyperMap	0.52	0.45	0.25	0.98	0.63	0.50	0.86	0.66	0.61	0.0
EBC-MCE	0.54	0.47	0.33	1.00	0.42	0.54	0.90	0.62	0.60	-1.6
RA-ncISO-EA	0.60	0.48	0.30	0.98	0.33	0.56	0.89	0.68	0.60	-1.6
RA-ISO-EA	0.79	0.44	0.14	0.96	0.35	0.53	0.89	0.69	0.60	-1.6
EBC-MCE-EA	0.82	0.44	0.19	0.95	0.29	0.56	0.91	0.61	0.60	-1.6
ISO-EA	0.57	0.43	0.19	0.99	0.44	0.56	0.90	0.69	0.60	-1.6
EBC-LE	0.50	0.45	0.29	1.00	0.58	0.51	0.89	0.54	0.60	-1.6
ncISO	0.50	0.41	0.20	0.94	0.60	0.50	0.89	0.70	0.59	-3.3
LE-EA	0.70	0.57	0.11	0.92	0.33	0.54	0.90	0.68	0.59	-3.3
EBC-ncMCE-EA	0.67	0.41	0.29	0.90	0.36	0.53	0.90	0.68	0.59	-3.3
RA-ncMCE	0.45	0.53	0.32	0.86	0.72	0.37	0.85	0.63	0.59	-3.3
HyperMapCN	0.48	0.46	0.38	0.87	0.61	0.42	0.82	0.67	0.59	-3.3
EBC-ISO-EA	0.66	0.38	0.33	0.95	0.29	0.53	0.90	0.68	0.59	-3.3
RA-LE-EA	0.62	0.53	0.26	0.96	0.22	0.52	0.89	0.68	0.59	-3.3
RA-MCE-EA	0.39	0.43	0.37	0.94	0.28	0.55	0.89	0.70	0.57	-6.6
ncMCE-EA	0.67	0.42	0.17	0.96	0.14	0.56	0.88	0.69	0.56	-8.2
ncMCE	0.58	0.59	0.23	0.78	0.32	0.42	0.87	0.62	0.55	-9.8
MCE-EA	0.56	0.24	0.29	0.92	0.19	0.56	0.88	0.71	0.55	-9.8
MCE	0.54	0.47	0.37	0.63	0.30	0.44	0.87	0.63	0.53	-13.1

Supplementary Table 3. Community detection on real networks with Label propagation algorithm. The table is equivalent to Supplementary Table 2, but the Label propagation algorithm is used rather than Infomap.

Method	Karate	Opsahl 8	Opsahl 9	Opsahl 10	Opsahl 11	Polbooks	Football	Polblogs	Mean	% Impr.
	N=34	N=43	N=44	N=77	N=77	N=105	N=115	N=1222		
	E=78	E=193	E=348	E=518	E=1088	E=441	E=613	E=16714		
	m=2.29	m=4.49	m=7.91	m=6.73	m=14.13	m=4.20	m=5.33	m=13.68		
	T=0.43	T=0.43	T=0.32	T=0.35	T=0.28	T=0.51	T=0.60	T=0.68		
	γ=2.12	γ=8.20	γ=5.92	γ=5.06	γ=4.87	γ=2.62	γ=9.09	γ=2.38		
	N_c=2	N_c=7	N_c=7	N_c=4	N_c=4	N_c=3	N_c=12	N_c=2		
RA-ISO-EA	0.58	0.44	0.53	1.00	0.85	0.56	0.89	0.65	0.69	+6.2
RA-MCE-EA	0.49	0.41	0.53	1.00	0.96	0.54	0.91	0.65	0.69	+6.2
ncISO-EA	0.68	0.44	0.47	1.00	0.89	0.54	0.86	0.67	0.69	+6.2
LPCS	0.57	0.55	0.41	1.00	0.96	0.54	0.89	0.63	0.69	+6.2
EBC-ncISO-EA	0.68	0.47	0.36	1.00	0.89	0.54	0.89	0.65	0.68	+4.6
EBC-MCE-EA	0.68	0.44	0.36	1.00	0.91	0.54	0.89	0.65	0.68	+4.6
EBC-ncMCE-EA	0.68	0.44	0.36	1.00	0.89	0.54	0.89	0.64	0.68	+4.6
ncMCE-EA	0.58	0.41	0.47	1.00	0.89	0.54	0.89	0.67	0.68	+4.6
EBC-MCE	0.49	0.51	0.36	1.00	0.96	0.54	0.89	0.67	0.68	+4.6
RA-ncMCE	0.53	0.55	0.46	0.91	1.00	0.48	0.89	0.61	0.68	+4.6
RA-LE-EA	0.58	0.53	0.36	1.00	0.90	0.54	0.91	0.64	0.68	+4.6
RA-ncMCE-EA	0.51	0.44	0.41	1.00	0.96	0.54	0.89	0.66	0.68	+4.6
EBC-ISO-EA	0.47	0.45	0.47	1.00	0.89	0.54	0.89	0.66	0.67	+3.1
EBC-LE-EA	0.68	0.44	0.36	0.90	0.89	0.54	0.86	0.66	0.67	+3.1
HyperMapCN	0.55	0.55	0.41	1.00	0.83	0.46	0.94	0.61	0.67	+3.1
MCE-EA	0.52	0.41	0.47	0.92	0.89	0.54	0.89	0.68	0.66	+1.5
EBC-ncISO	0.68	0.53	0.36	1.00	0.83	0.44	0.85	0.62	0.66	+1.5
LE	0.68	0.44	0.36	1.00	0.86	0.45	0.89	0.61	0.66	+1.5
unweighted	0.46	0.45	0.36	1.00	0.89	0.53	0.89	0.64	0.65	0.0
RA-ISO	0.68	0.44	0.36	1.00	0.73	0.54	0.85	0.63	0.65	0.0
ISO-EA	0.46	0.44	0.36	1.00	0.89	0.54	0.86	0.67	0.65	0.0
LE-EA	0.68	0.44	0.36	1.00	0.73	0.54	0.85	0.62	0.65	0.0
RA-ncISO-EA	0.50	0.44	0.36	1.00	0.85	0.54	0.85	0.66	0.65	0.0
ISO	0.52	0.44	0.36	0.96	0.86	0.54	0.86	0.63	0.65	0.0
RA-MCE	0.50	0.48	0.41	0.91	0.87	0.50	0.85	0.65	0.65	0.0
EBC-ISO	0.57	0.41	0.47	1.00	0.80	0.44	0.89	0.60	0.65	0.0
EBC-ncMCE	0.47	0.35	0.42	1.00	0.90	0.46	0.91	0.64	0.64	-1.5
ncISO	0.48	0.44	0.36	1.00	0.89	0.49	0.86	0.63	0.64	-1.5
RA-ncISO	0.50	0.48	0.36	1.00	0.76	0.54	0.85	0.62	0.64	-1.5
RA-LE	0.51	0.35	0.36	1.00	0.87	0.44	0.91	0.64	0.63	-3.1
EBC-LE	0.57	0.35	0.36	1.00	0.80	0.45	0.89	0.65	0.63	-3.1
HyperMap	0.45	0.56	0.36	0.96	0.78	0.43	0.87	0.62	0.63	-3.1
ncMCE	0.46	0.52	0.47	0.92	0.55	0.51	0.89	0.33	0.58	-10.8
MCE	0.42	0.42	0.47	0.92	0.56	0.47	0.91	0.33	0.56	-13.8

Supplementary Table 4. Community detection on real networks with Walktrap algorithm. The table is equivalent to Supplementary Table 2, but the Walktrap algorithm is used rather than Infomap.

Method	Karate	Opsahl 8	Opsahl 9	Opsahl 10	Opsahl 11	Polbooks	Football	Polblogs	Mean	% Impr.
	N=34	N=43	N=44	N=77	N=77	N=105	N=115	N=1222		
	E=78	E=193	E=348	E=518	E=1088	E=441	E=613	E=16714		
	m=2.29	m=4.49	m=7.91	m=6.73	m=14.13	m=4.20	m=5.33	m=13.68		
	T=0.43	T=0.43	T=0.32	T=0.35	T=0.28	T=0.51	T=0.60	T=0.68		
	γ=2.12	γ=8.20	γ=5.92	γ=5.06	γ=4.87	γ=2.62	γ=9.09	γ=2.38		
	N_c=2	N_c=7	N_c=7	N_c=4	N_c=4	N_c=3	N_c=12	N_c=2		
EBC-ISO-EA	0.78	0.50	0.41	1.00	0.96	0.45	0.90	0.63	0.70	+2.9
ISO-EA	0.58	0.50	0.46	1.00	1.00	0.51	0.93	0.66	0.70	+2.9
MCE-EA	0.51	0.55	0.53	0.96	0.96	0.54	0.90	0.66	0.70	+2.9
RA-ISO	0.68	0.55	0.39	1.00	0.96	0.49	0.90	0.62	0.70	+2.9
EBC-ncMCE	0.57	0.55	0.49	1.00	0.96	0.49	0.91	0.60	0.70	+2.9
EBC-ncISO-EA	0.56	0.55	0.41	1.00	0.93	0.52	0.90	0.62	0.69	+1.5
EBC-ncMCE-EA	0.47	0.55	0.41	1.00	0.96	0.54	0.93	0.62	0.69	+1.5
EBC-MCE-EA	0.57	0.55	0.41	1.00	0.96	0.51	0.91	0.62	0.69	+1.5
EBC-ISO	0.57	0.50	0.47	1.00	1.00	0.45	0.93	0.64	0.69	+1.5
EBC-ncISO	0.68	0.50	0.37	1.00	0.96	0.45	0.90	0.67	0.69	+1.5
EBC-LE-EA	0.68	0.51	0.42	0.96	0.92	0.52	0.87	0.63	0.69	+1.5
LE-EA	0.59	0.55	0.42	1.00	0.93	0.51	0.90	0.63	0.69	+1.5
LE	0.58	0.55	0.41	1.00	0.96	0.50	0.90	0.65	0.69	+1.5
EBC-LE	0.68	0.50	0.41	1.00	0.96	0.45	0.90	0.62	0.69	+1.5
unweighted	0.46	0.55	0.41	1.00	0.96	0.50	0.93	0.64	0.68	0.0
RA-ncISO-EA	0.49	0.55	0.41	1.00	0.96	0.52	0.90	0.63	0.68	0.0
ncISO-EA	0.52	0.46	0.46	1.00	0.96	0.48	0.91	0.66	0.68	0.0
EBC-MCE	0.47	0.49	0.49	1.00	0.96	0.54	0.90	0.59	0.68	0.0
RA-LE-EA	0.48	0.55	0.46	1.00	0.96	0.46	0.90	0.63	0.68	0.0
RA-ISO-EA	0.50	0.50	0.41	1.00	0.96	0.52	0.90	0.64	0.68	0.0
ISO	0.47	0.50	0.41	0.96	0.96	0.56	0.91	0.66	0.68	0.0
ncISO	0.47	0.52	0.42	1.00	0.96	0.47	0.91	0.67	0.68	0.0
RA-MCE-EA	0.49	0.52	0.42	1.00	0.92	0.52	0.91	0.64	0.68	0.0
RA-ncMCE-EA	0.47	0.55	0.42	1.00	0.96	0.46	0.91	0.64	0.68	0.0
ncMCE-EA	0.50	0.48	0.41	1.00	0.96	0.50	0.90	0.66	0.68	0.0
RA-LE	0.55	0.48	0.41	1.00	0.92	0.45	0.89	0.65	0.67	-1.5
RA-ncISO	0.47	0.48	0.41	1.00	0.92	0.47	0.90	0.63	0.66	-2.9
RA-MCE	0.47	0.54	0.42	0.93	0.92	0.51	0.86	0.63	0.66	-2.9
RA-ncMCE	0.47	0.60	0.41	0.93	1.00	0.46	0.88	0.48	0.66	-2.9
LPCS	0.47	0.50	0.41	1.00	0.96	0.46	0.89	0.61	0.66	-2.9
HyperMapCN	0.55	0.49	0.41	1.00	0.88	0.46	0.90	0.60	0.66	-2.9
HyperMap	0.56	0.57	0.32	1.00	0.92	0.47	0.87	0.57	0.66	-2.9
MCE	0.52	0.46	0.34	0.75	0.56	0.47	0.88	0.54	0.56	-17.6
ncMCE	0.54	0.49	0.34	0.82	0.40	0.43	0.89	0.55	0.56	-17.6

Supplementary Table 5. Community detection on real networks with Louvain algorithm (HD-network). The table is equivalent to Supplementary Table 1, but the HD-weighted network (see Equation 13) is given in input to the Louvain algorithm rather than the HSP-kernel (see Equation 14).

Method	AS 201501 IPv6	AS 200909 IPv4	AS 200912 IPv4	AS 201003 IPv4	AS 201006 IPv4	AS 201009 IPv4	AS 201012 IPv4	AS 201501 IPv4	Mean	% Impr.
	N=5143	N=24091	N=25910	N=26307	N=26756	N=28353	N=29333	N=37542		
	E=13446	E=59531	E=63435	E=66089	E=68150	E=73722	E=78054	E=95019		
	m=2.61	m=2.47	m=2.45	m=2.51	m=2.55	m=2.60	m=2.66	m=2.53		
	T=0.65	T=0.64	T=0.64	T=0.63	T=0.63	T=0.63	T=0.62	T=0.64		
	γ=2.30	γ=2.12	γ=2.11	γ=2.26	γ=2.08	γ=2.23	γ=2.22	γ=2.07		
	N_c=151	N_c=203	N_c=206	N_c=204	N_c=204	N_c=208	N_c=212	N_c=222		
EBC-ncISO-EA	0.54	0.61	0.59	0.60	0.60	0.62	0.59	0.62	0.60	+3.4
EBC-ISO-EA	0.53	0.61	0.60	0.59	0.59	0.62	0.61	0.62	0.60	+3.4
EBC-ISO	0.52	0.61	0.60	0.61	0.61	0.62	0.62	0.62	0.60	+3.4
EBC-ncISO	0.53	0.61	0.60	0.60	0.59	0.62	0.62	0.60	0.60	+3.4
EBC-MCE	0.54	0.61	0.60	0.61	0.61	0.60	0.60	0.62	0.60	+3.4
EBC-ncMCE-EA	0.54	0.61	0.59	0.60	0.60	0.61	0.60	0.59	0.59	+1.7
RA-ncMCE-EA	0.56	0.60	0.59	0.59	0.59	0.59	0.59	0.59	0.59	+1.7
EBC-MCE-EA	0.54	0.60	0.59	0.61	0.61	0.60	0.60	0.58	0.59	+1.7
RA-MCE-EA	0.57	0.60	0.59	0.59	0.58	0.59	0.59	0.59	0.59	+1.7
EBC-LE-EA	0.55	0.58	0.60	0.60	0.59	0.60	0.60	0.58	0.59	+1.7
RA-LE-EA	0.54	0.61	0.58	0.59	0.60	0.60	0.59	0.59	0.59	+1.7
EBC-ncMCE	0.53	0.61	0.59	0.60	0.61	0.59	0.60	0.61	0.59	+1.7
RA-ncMCE	0.57	0.60	0.59	0.59	0.59	0.59	0.59	0.59	0.59	+1.7
RA-MCE	0.56	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	+1.7
MCE-EA	0.55	0.60	0.59	0.59	0.60	0.59	0.59	0.59	0.59	+1.7
unweighted	0.56	0.58	0.58	0.58	0.58	0.59	0.59	0.58	0.58	0.0
RA-ISO-EA	0.55	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.58	0.0
RA-ncISO	0.55	0.60	0.58	0.58	0.59	0.59	0.60	0.58	0.58	0.0
EBC-LE	0.55	0.57	0.59	0.60	0.59	0.62	0.57	0.57	0.58	0.0
RA-LE	0.55	0.57	0.58	0.59	0.60	0.62	0.57	0.59	0.58	0.0
RA-ISO	0.55	0.59	0.58	0.59	0.59	0.59	0.59	0.59	0.58	0.0
RA-ncISO-EA	0.55	0.59	0.58	0.59	0.58	0.59	0.59	0.58	0.58	0.0
LE-EA	0.55	0.60	0.58	0.58	0.58	0.59	0.58	0.58	0.58	0.0
ncMCE-EA	0.55	0.59	0.58	0.59	0.58	0.59	0.59	0.58	0.58	0.0
MCE	0.55	0.59	0.58	0.58	0.58	0.59	0.59	0.58	0.58	0.0
ncISO	0.53	0.59	0.58	0.58	0.58	0.60	0.60	0.58	0.58	0.0
ncMCE	0.55	0.59	0.58	0.57	0.58	0.59	0.58	0.59	0.58	0.0
ncISO-EA	0.53	0.59	0.58	0.58	0.58	0.60	0.59	0.58	0.58	0.0
LE	0.55	0.58	0.58	0.59	0.58	0.58	0.58	0.59	0.58	0.0
ISO	0.52	0.59	0.58	0.58	0.58	0.58	0.59	0.58	0.57	-1.7
ISO-EA	0.53	0.58	0.57	0.58	0.58	0.58	0.59	0.57	0.57	-1.7
LPCS	0.56	0.57	0.57	0.58	0.58	0.57	0.58	0.57	0.57	-1.7

Supplementary Table 6. Community detection on Internet networks with Infomap algorithm. The table is equivalent to Supplementary Table 2, but the community detection is performed on the Internet networks rather than on the small-size real networks.

Method	AS 201501 IPv6	AS 200909 IPv4	AS 200912 IPv4	AS 201003 IPv4	AS 201006 IPv4	AS 201009 IPv4	AS 201012 IPv4	AS 201501 IPv4	Mean	% Impr.
	N=5143	N=24091	N=25910	N=26307	N=26756	N=28353	N=29333	N=37542		
	E=13446	E=59531	E=63435	E=66089	E=68150	E=73722	E=78054	E=95019		
	m=2.61	m=2.47	m=2.45	m=2.51	m=2.55	m=2.60	m=2.66	m=2.53		
	T=0.65	T=0.64	T=0.64	T=0.63	T=0.63	T=0.63	T=0.62	T=0.64		
	γ=2.30	γ=2.12	γ=2.11	γ=2.26	γ=2.08	γ=2.23	γ=2.22	γ=2.07		
	N_c=151	N_c=203	N_c=206	N_c=204	N_c=204	N_c=208	N_c=212	N_c=222		
LPCS	0.51	0.55	0.51	0.56	0.52	0.54	0.46	0.32	0.50	+117.4
ncMCE	0.32	0.44	0.36	0.55	0.28	0.28	0.43	0.21	0.36	+56.5
RA-MCE	0.21	0.26	0.50	0.36	0.29	0.47	0.24	0.32	0.33	+43.5
EBC-LE	0.36	0.44	0.35	0.26	0.19	0.28	0.37	0.34	0.33	+43.5
LE	0.15	0.40	0.43	0.40	0.34	0.33	0.26	0.27	0.32	+39.1
RA-ncISO	0.30	0.37	0.37	0.42	0.27	0.32	0.27	0.18	0.31	+34.8
MCE	0.27	0.38	0.35	0.39	0.27	0.26	0.26	0.24	0.30	+30.4
RA-ncMCE	0.20	0.33	0.30	0.29	0.51	0.22	0.39	0.19	0.30	+30.4
LE-EA	0.14	0.40	0.39	0.32	0.33	0.28	0.27	0.23	0.29	+26.1
RA-LE	0.35	0.44	0.27	0.27	0.17	0.31	0.38	0.17	0.29	+26.1
ncMCE-EA	0.32	0.35	0.31	0.36	0.23	0.23	0.30	0.21	0.29	+26.1
MCE-EA	0.19	0.35	0.31	0.34	0.26	0.24	0.24	0.18	0.27	+17.4
RA-ISO	0.12	0.29	0.36	0.27	0.25	0.32	0.23	0.16	0.25	+8.7
EBC-LE-EA	0.13	0.39	0.28	0.28	0.21	0.23	0.22	0.20	0.24	+4.3
RA-MCE-EA	0.22	0.29	0.32	0.26	0.22	0.23	0.23	0.17	0.24	+4.3
unweighted	0.07	0.30	0.41	0.29	0.29	0.19	0.19	0.15	0.23	0.0
RA-ncMCE-EA	0.16	0.23	0.25	0.25	0.36	0.22	0.28	0.14	0.23	0.0
EBC-ISO-EA	0.15	0.23	0.32	0.25	0.21	0.27	0.27	0.18	0.23	0.0
ncISO	0.08	0.24	0.31	0.29	0.26	0.27	0.19	0.17	0.23	0.0
EBC-ncMCE-EA	0.10	0.24	0.25	0.27	0.22	0.24	0.22	0.19	0.22	-4.3
EBC-ncISO-EA	0.12	0.23	0.26	0.25	0.23	0.25	0.24	0.19	0.22	-4.3
EBC-ncISO	0.18	0.22	0.22	0.25	0.25	0.22	0.22	0.19	0.22	-4.3
RA-LE-EA	0.13	0.29	0.24	0.24	0.19	0.20	0.20	0.19	0.21	-8.7
ISO	0.09	0.30	0.29	0.25	0.18	0.21	0.19	0.16	0.21	-8.7
EBC-ISO	0.12	0.29	0.24	0.22	0.20	0.20	0.20	0.19	0.21	-8.7
ISO-EA	0.10	0.30	0.27	0.26	0.17	0.22	0.19	0.15	0.21	-8.7
EBC-MCE-EA	0.10	0.21	0.23	0.24	0.19	0.26	0.20	0.16	0.20	-13.0
RA-ncISO-EA	0.13	0.30	0.24	0.27	0.14	0.20	0.17	0.16	0.20	-13.0
ncISO-EA	0.08	0.21	0.36	0.26	0.21	0.21	0.17	0.13	0.20	-13.0
RA-ISO-EA	0.11	0.21	0.29	0.25	0.23	0.24	0.18	0.14	0.20	-13.0
EBC-MCE	0.07	0.21	0.26	0.22	0.22	0.25	0.19	0.16	0.20	-13.0
EBC-ncMCE	0.08	0.24	0.25	0.21	0.23	0.19	0.16	0.19	0.19	-17.4

Supplementary Table 7. Community detection on Internet networks with Label propagation algorithm. The table is equivalent to Supplementary Table 6, but the Label propagation algorithm is used rather than Infomap.

Method	AS 201501 IPv6	AS 200909 IPv4	AS 200912 IPv4	AS 201003 IPv4	AS 201006 IPv4	AS 201009 IPv4	AS 201012 IPv4	AS 201501 IPv4	Mean	% Impr.
	N=5143	N=24091	N=25910	N=26307	N=26756	N=28353	N=29333	N=37542		
	E=13446	E=59531	E=63435	E=66089	E=68150	E=73722	E=78054	E=95019		
	m=2.61	m=2.47	m=2.45	m=2.51	m=2.55	m=2.60	m=2.66	m=2.53		
	T=0.65	T=0.64	T=0.64	T=0.63	T=0.63	T=0.63	T=0.62	T=0.64		
	γ=2.30	γ=2.12	γ=2.11	γ=2.26	γ=2.08	γ=2.23	γ=2.22	γ=2.07		
	N_c=151	N_c=203	N_c=206	N_c=204	N_c=204	N_c=208	N_c=212	N_c=222		
RA-MCE	0.52	0.62	0.66	0.63	0.64	0.65	0.64	0.64	0.63	+1.6
unweighted	0.51	0.65	0.64	0.63	0.63	0.64	0.65	0.63	0.62	0.0
RA-MCE-EA	0.52	0.63	0.65	0.62	0.63	0.65	0.65	0.64	0.62	0.0
RA-ncISO-EA	0.51	0.63	0.65	0.64	0.63	0.66	0.64	0.63	0.62	0.0
EBC-ncISO-EA	0.51	0.65	0.64	0.62	0.64	0.64	0.65	0.63	0.62	0.0
EBC-MCE-EA	0.48	0.63	0.65	0.62	0.63	0.65	0.64	0.61	0.62	0.0
RA-ncMCE-EA	0.51	0.64	0.65	0.64	0.60	0.64	0.65	0.63	0.62	0.0
EBC-ISO-EA	0.48	0.63	0.61	0.64	0.65	0.65	0.65	0.61	0.62	0.0
RA-ISO-EA	0.49	0.64	0.65	0.65	0.61	0.64	0.64	0.63	0.62	0.0
EBC-ncISO	0.48	0.64	0.65	0.65	0.64	0.65	0.65	0.61	0.62	0.0
RA-ncMCE	0.51	0.63	0.64	0.61	0.65	0.64	0.64	0.65	0.62	0.0
MCE	0.51	0.64	0.64	0.65	0.63	0.63	0.64	0.62	0.62	0.0
EBC-ISO	0.46	0.64	0.65	0.64	0.64	0.66	0.65	0.62	0.62	0.0
LE-EA	0.49	0.64	0.63	0.65	0.62	0.65	0.64	0.64	0.62	0.0
ISO	0.48	0.65	0.64	0.64	0.63	0.64	0.64	0.62	0.62	0.0
ncISO-EA	0.49	0.64	0.64	0.64	0.63	0.61	0.64	0.64	0.62	0.0
LE	0.50	0.67	0.64	0.62	0.62	0.65	0.64	0.62	0.62	0.0
MCE-EA	0.51	0.63	0.63	0.64	0.63	0.64	0.65	0.62	0.62	0.0
ncISO	0.47	0.63	0.64	0.64	0.63	0.63	0.64	0.64	0.62	0.0
LPCS	0.53	0.63	0.64	0.63	0.63	0.65	0.62	0.63	0.62	0.0
EBC-ncMCE-EA	0.47	0.61	0.65	0.63	0.63	0.65	0.64	0.62	0.61	-1.6
ncMCE-EA	0.51	0.63	0.63	0.64	0.63	0.64	0.61	0.63	0.61	-1.6
ISO-EA	0.49	0.64	0.64	0.63	0.62	0.64	0.64	0.62	0.61	-1.6
EBC-LE-EA	0.49	0.63	0.63	0.63	0.62	0.64	0.64	0.61	0.61	-1.6
RA-LE-EA	0.48	0.63	0.64	0.63	0.62	0.64	0.63	0.62	0.61	-1.6
EBC-LE	0.53	0.62	0.64	0.62	0.63	0.64	0.61	0.61	0.61	-1.6
RA-LE	0.53	0.61	0.62	0.64	0.62	0.63	0.61	0.62	0.61	-1.6
EBC-MCE	0.45	0.62	0.65	0.61	0.62	0.65	0.64	0.63	0.61	-1.6
EBC-ncMCE	0.44	0.65	0.65	0.61	0.61	0.64	0.64	0.62	0.61	-1.6
RA-ISO	0.48	0.60	0.61	0.65	0.62	0.65	0.62	0.62	0.61	-1.6
RA-ncISO	0.52	0.60	0.63	0.62	0.63	0.63	0.63	0.60	0.61	-1.6
ncMCE	0.51	0.62	0.62	0.61	0.62	0.64	0.61	0.63	0.61	-1.6

Supplementary Table 8. Community detection on Internet networks with Walktrap algorithm. The table is equivalent to Supplementary Table 6, but the Walktrap algorithm is used rather than Infomap.

Method	Mean GR-score 2D	Mean GR-score 3D	Improvement	p-value
ISO	0.61	0.67	+0.06	**< 0.001**
ncISO	0.62	0.66	+0.04	**< 0.001**
RA-ISO	0.64	0.65	+0.01	0.296
EBC-ISO	0.61	0.61	0.00	0.494
EBC-ncISO	0.62	0.62	0.00	0.499
RA-ncISO	0.65	0.65	0.00	0.429
LE	0.67	0.66	-0.01	0.296
RA-LE	0.68	0.67	-0.01	0.259
EBC-LE	0.68	0.66	-0.02	0.113

Supplementary Table 9. Comparison of 2D and 3D greedy routing on PSO synthetic networks. The same PSO networks considered in Fig. 3 have been mapped both in 2D and 3D using the manifold-based coalescent embedding techniques and the greedy routing in the hyperbolic space has been evaluated. The table reports for each method the mean GR-score over all the PSO parameter combinations, both in 2D and in 3D, highlighting the 3D-improvement. The GR-score is a metric to evaluate the efficiency of the greedy routing, which assumes values between 0, when all the routings are unsuccessful, and 1, when all the packets reach the destination through the shortest path (see Methods for details). The rightmost column shows for each method the p-value of the permutation test for the mean (10000 iterations) performed considering the two vectors of GR-scores related to 2D and 3D. The p-values lower than the significance level of 0.05 are highlighted in bold.

GR-score 2D

Method	Karate	Opsahl 8	Opsahl 9	Opsahl 10	Opsahl 11	Polbooks	Football	Polblogs	Mean
RA-ncISO	0.91	0.92	0.95	0.92	0.96	0.47	0.65	0.41	0.77
RA-LE	0.85	0.91	0.97	0.91	0.98	0.49	0.70	0.30	0.76
RA-ISO	0.86	0.93	0.93	0.90	0.96	0.43	0.62	0.43	0.76
ncISO	0.82	0.94	0.95	0.91	0.97	0.40	0.60	0.44	0.75
ISO	0.79	0.93	0.91	0.88	0.98	0.40	0.56	0.51	0.75
LE	0.80	0.93	0.97	0.93	0.94	0.42	0.59	0.38	0.74
EBC-LE	0.78	0.92	0.95	0.93	0.91	0.40	0.72	0.26	0.73
EBC-ncISO	0.80	0.83	0.94	0.91	0.92	0.40	0.58	0.45	0.73
EBC-ISO	0.84	0.89	0.94	0.85	0.94	0.36	0.57	0.38	0.72

GR-score 3D

Method	Karate	Opsahl 8	Opsahl 9	Opsahl 10	Opsahl 11	Polbooks	Football	Polblogs	Mean
RA-ISO	0.86	0.97	0.97	0.96	0.94	0.44	0.77	0.47	0.80
RA-ncISO	0.80	0.96	0.97	0.96	0.94	0.44	0.76	0.46	0.79
ncISO	0.76	0.97	0.97	0.97	0.94	0.44	0.67	0.49	0.78
LE	0.75	0.91	0.98	0.94	0.94	0.47	0.75	0.46	0.77
RA-LE	0.75	0.89	0.98	0.96	0.94	0.51	0.75	0.38	0.77
EBC-ISO	0.80	0.92	0.95	0.93	0.92	0.41	0.76	0.46	0.77
EBC-ncISO	0.76	0.93	0.96	0.92	0.92	0.41	0.78	0.45	0.77
ISO	0.72	0.94	0.97	0.96	0.94	0.44	0.65	0.51	0.77
EBC-LE	0.79	0.88	0.92	0.95	0.93	0.43	0.74	0.11	0.72

Improvement

Method	Karate	Opsahl 8	Opsahl 9	Opsahl 10	Opsahl 11	Polbooks	Football	Polblogs	Mean	p-value
EBC-ISO	-0.04	0.03	0.01	0.08	-0.02	0.05	0.19	0.08	+0.05	0.409
EBC-ncISO	-0.04	0.10	0.02	0.01	0.00	0.01	0.20	0.00	+0.04	0.412
RA-ISO	0.00	0.04	0.04	0.06	-0.02	0.01	0.15	0.04	+0.04	0.359
LE	-0.05	-0.02	0.01	0.01	0.00	0.05	0.16	0.08	+0.03	0.376
ncISO	-0.06	0.03	0.02	0.06	-0.03	0.04	0.07	0.05	+0.03	0.434
ISO	-0.07	0.01	0.06	0.08	-0.04	0.04	0.09	0.00	+0.02	0.424
RA-ncISO	-0.11	0.04	0.02	0.04	-0.02	-0.03	0.11	0.05	+0.02	0.451
RA-LE	-0.10	-0.02	0.01	0.05	-0.04	0.02	0.05	0.08	+0.01	0.421
EBC-LE	0.01	-0.04	-0.03	0.02	0.02	0.03	0.02	-0.15	-0.01	0.493

Supplementary Table 10. Comparison of 2D and 3D greedy routing on real networks.
The 8 real networks whose statistics are reported in Table 1 have been mapped both in 2D and 3D using the manifold-based coalescent embedding techniques and the greedy routing in the hyperbolic space has been evaluated. The table reports for each method and for each network the GR-score both in 2D and in 3D, in addition to the 3D-improvement. The GR-score is a metric to evaluate the efficiency of the greedy routing, which assumes values between 0, when all the routings are unsuccessful, and 1, when all the packets reach the destination through the shortest path (see Methods for details). The rightmost column shows for each method the p-value of the permutation test for the mean (10000 iterations) performed considering the two vectors of GR-scores related to 2D and 3D.

	GR-score 2D		GR-score 3D		Improvement		
Method	AS 201501 IPv6	AS 200909 IPv4	AS 201501 IPv6	AS 200909 IPv4	AS 201501 IPv6	AS 200909 IPv4	Mean
RA-LE	0.02	0.01	0.02	0.01	0.00	0.00	0.00
RA-ISO	0.15	0.10	0.10	0.04	-0.05	-0.06	-0.06
EBC-ncISO	0.15	0.12	0.10	0.05	-0.05	-0.07	-0.06
EBC-ISO	0.17	0.11	0.11	0.04	-0.06	-0.07	-0.07
RA-ncISO	0.16	0.14	0.10	0.04	-0.06	-0.10	-0.08
EBC-LE	0.02	0.34	0.02	0.01	0.00	-0.33	-0.17
LE	0.23	0.28	0.13	0.05	-0.10	-0.23	-0.17
ncISO	0.32	0.24	0.16	0.06	-0.16	-0.18	-0.17
ISO	0.36	0.30	0.17	0.06	-0.19	-0.24	-0.22

Supplementary Table 11. Comparison of 2D and 3D greedy routing on Internet networks.
Two of the Internet networks whose statistics are reported in Supplementary Table 6 have been mapped both in 2D and 3D using the manifold-based coalescent embedding techniques and the greedy routing in the hyperbolic space has been evaluated. The table reports for each method and for each network the GR-score both in 2D and in 3D, in addition to the 3D-improvement. The GR-score is a metric to evaluate the efficiency of the greedy routing, which assumes values between 0, when all the routings are unsuccessful, and 1, when all the packets reach the destination through the shortest path (see Methods for details). Differently from the previous tables, the statistical test has not been performed due to the reduced number of networks considered.

Louvain					Infomap				
Method	Mean NMI 2D	Mean NMI 3D	Impr.	p-value	Method	Mean NMI 2D	Mean NMI 3D	Impr.	p-value
RA-LE	0.66	0.69	+0.03	0.380	**ISO**	0.67	0.72	+0.05	0.343
LE	0.68	0.70	+0.02	0.452	**RA-LE**	0.65	0.67	+0.02	0.429
ISO	0.67	0.69	+0.02	0.413	**EBC-ISO**	0.69	0.71	+0.02	0.387
ncISO	0.66	0.67	+0.01	0.482	**ncISO**	0.67	0.69	+0.02	0.404
EBC-LE	0.64	0.65	+0.01	0.481	**EBC-ncISO**	0.70	0.71	+0.01	0.424
EBC-ncISO	0.69	0.70	+0.01	0.450	**LE**	0.68	0.69	+0.01	0.494
RA-ISO	0.69	0.69	0.00	0.487	**RA-ncISO**	0.68	0.68	0.00	0.489
EBC-ISO	0.71	0.70	-0.01	0.473	**RA-ISO**	0.70	0.69	-0.01	0.441
RA-ncISO	0.70	0.68	-0.02	0.453	**EBC-LE**	0.68	0.66	-0.02	0.425
Label Propagation					**Walktrap**				
Method	Mean NMI 2D	Mean NMI 3D	Impr.	p-value	Method	Mean NMI 2D	Mean NMI 3D	Impr.	p-value
ncISO	0.59	0.62	+0.03	0.410	**EBC-LE**	0.63	0.66	+0.03	0.436
EBC-LE	0.60	0.63	+0.03	0.391	**RA-LE**	0.63	0.64	+0.01	0.492
RA-ISO	0.63	0.65	+0.02	0.430	**EBC-ncISO**	0.66	0.67	+0.01	0.469
LE	0.62	0.62	0.00	0.491	**ISO**	0.65	0.65	0.00	0.493
RA-LE	0.63	0.63	0.00	0.488	**ncISO**	0.64	0.64	0.00	0.505
RA-ncISO	0.63	0.62	-0.01	0.471	**RA-ncISO**	0.64	0.64	0.00	0.503
EBC-ISO	0.63	0.61	-0.02	0.438	**EBC-ISO**	0.65	0.65	0.00	0.483
EBC-ncISO	0.64	0.61	-0.03	0.419	**LE**	0.66	0.65	-0.01	0.474
ISO	0.67	0.63	-0.04	0.360	**RA-ISO**	0.65	0.64	-0.01	0.456

Supplementary Table 12. Comparison of 2D and 3D community detection on real networks. The 8 real networks whose statistics are reported in Table 1 have been mapped both in 2D and 3D using the manifold-based coalescent embedding techniques and the community detection has been evaluated exploiting the 2D and 3D hyperbolic distances to weight the input matrix for the four community detection algorithms. The table reports for each method the mean NMI over all the networks both in 2D and in 3D, highlighting the 3D-improvement. NMI is the normalized mutual information and represents the shared information between two distributions, normalized between 0 and 1, where 1 indicates that the communities detected by the algorithm perfectly correspond to the ground truth communities (see Methods for details). The rightmost column of each community detection algorithm shows for each embedding method the p-value of the permutation test for the mean (10000 iterations) performed considering the two vectors of GR-scores related to 2D and 3D.

Supplementary Notes

Supplementary Note 1: Repulsion-Attraction pre-weighting rules

$$x_{ij}^{RA1} = \frac{d_i + d_j + d_i d_j}{1 + CN_{ij}} \tag{1}$$

$$x_{ij}^{RA2} = \frac{1 + e_i + e_j + e_i e_j}{1 + CN_{ij}} \tag{2}$$

$$x_{ij}^{RA3} = \frac{d_i + d_j}{1 + CN_{ij}} \tag{3}$$

$$x_{ij}^{RA4} = \frac{d_i d_j}{1 + CN_{ij}} \tag{4}$$

In the mathematical expressions: x_{ij} is the value of an edge (i, j) in the adjacency matrix; d_i is the degree of the node *i*; e_i is the external degree of the node *i* with respect to node *j* (links to nodes that are neither common neighbours nor the node *j*); CN_{ij} are the common neighbours of nodes *i* and *j*.

Supplementary Discussion

Greedy routing performance in synthetic and real networks

In this section, we would like to explain the reason why RA-ncMCE resulted the best performing among the coalescent embedding methods on the greedy routing tests.

Firstly, we would like to underline the relevant increase of performance obtained by ncMCE when the RA pre-weighting is applied, which confirms the efficacy of the RA rule. The RA pre-weighting is very effective to suggest the hidden geometry to extract the MST on the basis of which the distances that are collected in the MC-kernel are approximated (see Methods for details). In fact, ncMCE alone offers a very poor performance in GR, while RA-ncMCE provides top performance between the coalescent embedding techniques.

Secondly, the fact that ncMCE is performing better than MCE in problems of network embedding was also proved and discussed in a previous publication [36], and it is related with the effect of the kernel centring procedure, therefore we will not discuss further here.

At last, a theoretical digression is necessary in order to explain why RA-ncMCE performed better than the manifold-based coalescent embedding techniques. The embedding methods based on matrix factorization are all global methods because they exploit an *NxN* matrix decomposition [111]. However, they mainly work in different ways. LE is a neighbourhood-preserving global method, in fact, the Laplacian matrix to which eigen-decomposition is applied only contains information about connected nodes, therefore it infers angular coordinates that give preference to put connected nodes closer. ISO and ncISO, on the contrary, belong to the class of global methods that aim to preserve the global topology of the neighbourhood graph that approximates the hidden manifold geometry, therefore they do not concentrate exclusively on the accurate preservation of connected points at close angular coordinates. Indeed, they apply singular value decomposition to a distance kernel, which contains information about both connected and disconnected nodes. It means that they attempt to preserve geometry at all scales, therefore introducing and distributing the error at all the scales. ISO procedure does not give preference to an accurate preservation of connected points that is a necessary procedure for effective GR but, on the other hand, tries to fulfil the second important condition to put disconnected points far in the angular coordinates. ncMCE is a global method that preserves a locally-reconstructed (by means of MST) global geometry by overestimating distances between disconnected nodes. Since the MC-kernel is obtained by computing all pairwise transversal distances over the MST, in practice distances for both connected and disconnected nodes are approximated. However, the distances of connected nodes will be 'fairly' estimated, while the ones of disconnected nodes will be overrated. In conclusion, minimum curvilinearity, which is the mechanism of generation of the MC-kernel, favours inference of kernel distances that preserve connected nodes close in the angular coordinates, and push disconnected nodes far apart in the angular coordinates.

Community detection on real networks

This section is intended to provide further discussions about the results on the community detection application. As first, we would like to comment the fact that, differently from the embedding evaluation on the synthetic networks, in community detection on real networks ncISO-based and MCE-based coalescent embedding techniques are significantly better than LE-based methods (Table 1-2 and Supplementary Table 1-4). Not to be overlooked, EBC-ncISO-EA is the only method that improves with respect to all the four unweighted algorithms. As expected, this finding suggests that the results obtained on synthetic networks are indicative but should be taken with caution. Real networks might have a geometry that is even more tree-like and hyperbolic than the one hypothesized by the PSO model (for this reason MCE-based techniques can perform better on real networks), and although the topology of real networks is certainly conditioned by the hyperbolic geometry this is however one of the factors that shape their structure. On the other side, good results are achieved also for networks with out of range γ values such as Opsahl_11. Since it has been demonstrated that a scale-free degree distribution is a necessary condition for hyperbolic geometry [17], this result demonstrates that the methods can reach good performances also for networks whose latent geometry might be weakly hyperbolic.

As second point, we want to highlight that also in community detection simulations on real world networks the contribution offered by EA is evident. Except for Label propagation where the EA and non-EA methods show a mixed ranking (Supplementary Table 3), in Louvain, Infomap and Walktrap the EA-based coalescent embedding techniques offer the best performance (Supplementary Table 1, 2 and 4), confirming that the adjustment of local embedding uncertainty can be crucial for effective coalescent embedding also in real applications. We note that Louvain was the only method for which the improvement in community detection was higher using not only the geometrical information between the connected nodes but also between disconnected nodes. Supplementary Table 1 and 5 show the different results using either a kernel or a weighted network in input. For the other community detection algorithms the results are not reported since the usage of a kernel led to totally wrong predictions.

In order to test the coalescent embedding methods on real networks of larger size, the community detection has been performed also on Internet networks ranging from 5000 up to 37000 nodes, where each node represents an Autonomous System and the connections indicate the IPv4 or IPv6 topology (see dataset description for details). For Louvain, the processing of the kernel matrix for large networks resulted to be too computationally expensive from the point of view of the memory requirements, therefore the results are not reported. For Infomap and Walktrap, most of the embedding methods obtained the same performance as the unweighted variant, and also the other techniques did not show a big deviation from that reference (Supplementary Table 6 and 8). However, for Infomap, many MCE-based approaches offered a small increase of performance and the only improving method for Walktrap is also MCE-based. Differently, for Label propagation the unweighted variant obtained a very low result, therefore there was a higher margin of improvement and the usage of the geometrical information led to a significant boosting (Supplementary Table 7).

Beyond the two-dimensional space

Before starting with the analysis of the results, there is the need to discuss a preliminary point.

Coalescent embedding at the moment includes two different types, the manifold-based (LE, ISO, ncISO) and the Minimum Curvilinearity (MCE, ncMCE) approaches. The latter ones are fundamentally different, since they learn the nonlinear similarities by means of MST and linearize the hidden pattern providing a hierarchical-based mapping. Therefore, on one side we are seeking to exploit an additional dimension of embedding, on the other side the power of the Minimum Curvilinearity methods is the compression of information in a single dimension: it is evident by definition that it would be a contradiction to adopt them for this investigation. Furthermore, while the rearrangement of the linearized similarities over a circumference remains intuitive, it is not trivial to find a meaningful way for reorganizing the linearized pattern over a sphere. Note that, for analogous reasons, also the equidistant adjustment is not adopted. On the contrary, the manifold-based approaches offer less compression capabilities and therefore do not exclude the presence of potentially useful information in the third dimension. Moreover, since the hidden similarity pattern can remain nonlinear also in the embedded space, the similarities can be directly accommodated to the sphere without the need of any particular reorganization, as it happens using two dimensions.

Supplementary Table 9 reports the mean difference between the GR scores of the 3D versus 2D greedy routing performed on the PSO networks embedded in the hyperbolic space, where the mean is taken over all the parameter combinations. The table highlights a small even though significant (p-value < 0.001) improvement obtained with the addition of the third dimension in ISO and ncISO. However, there is no significant improvement for the other methods, and for the LE-based approaches there is even a small decline. Supplementary Table 10 reports the mean difference between the GR scores of the 3D versus 2D greedy routing performed on the real small-size networks embedded in the hyperbolic space, where the mean is taken over all the networks. The table shows that, except for EBC-LE, the methods slightly improve the performance with a gain in NMI up to 0.05, but not significantly. Supplementary Table 11 reports the difference between the GR scores of the 3D versus 2D greedy routing performed on two AS networks embedded in the hyperbolic space, one middle size and one large size. The table underlines the presence of a general performance decrease with the addition of the third dimension.

Supplementary Table 12 reports the mean difference between the NMI scores of the 3D versus 2D community detection performed on the real small-size networks embedded in the hyperbolic space, where the mean is taken over all the networks. The table highlights that for each community detection method the coalescent embedding approaches oscillate between a small increase and a little decrease of performance using the third dimension, but the difference is not significant in any case.

Notes on pre-weighting, rich-clubness and angular adjustment

The proposed class of coalescent embedding algorithms includes several variants and, except for the different machine learning techniques, the variations are given by the pre-weighting and the angular adjustment, whose contribution will be now discussed.

Out of question is the positive effect of the pre-weighting on the embedding accuracy. All the simulations highlighted that suggesting topological similarities between the connected nodes makes the inference of the coordinates more precise and leads to remarkable improvements in performance (Supplementary Fig. 2-6). Furthermore, both the local-based RA rule and the global-based EBC rule resulted to be effective. Since there was not a unique possible mathematical formulation of the RA formula, we tested four variants differing in the way in which the degrees of the connected nodes are combined, the mathematical expressions are shown in Supplementary Note 1 and the results in Supplementary Fig. 23. The variant RA_1 gave the best results for the embedding of small-size PSO networks, whereas the variant RA_2 for the large-size PSO networks, although the performance is very similar for all of them. Therefore, we here propose to adopt both the versions RA_1 and RA_2, as reported in Fig. 2. Since in most of the simulations the networks are small and the results of the two variants are very close, for sake of brevity we showed all the other results only for RA_1. We let notice that from a theoretical point of view the formula RA_2 is more correct, in fact it conceptually splits the neighbours of the adjacent nodes in two non-overlapping subsets: the neighbours not in common (external degree), responsible for the repulsive part, and the common neighbours, determining the attractive part. In the other formulas, instead, the numerator considers not the external degree but the degree of the adjacent nodes, which includes also the common neighbours. However, on the tested networks it emerges that this conceptual difference does not lead to a substantial performance improvement, but it might play an important role for networks of larger size.

It might be argued that the repulsion between high (external) degree nodes implied by the RA rule is in contrast with the existence of rich-clubs. In rich-club networks, high degree nodes (hubs) tend to connect each other [112]. However, we would like to clarify that the repulsive part of the rule is not suggesting that nodes with high (external) degree tend to be disconnected. It suggests that they tend to dominate geometrically distant regions, which does not exclude their connectivity and therefore it should not be theoretically in contrast to the existence of rich-clubs. In order to prove this point by experiments, we started performing a statistical test for rich-clubness[110] on the PSO networks used for the previous simulations, the p-values are reported in Supplementary Fig. 25. The statistical test highlights that for most of the parameter combinations, in particular for $m = [4, 6]$ and $N = [500, 1000]$ the networks present a significant rich-club, whereas for more sparse ($m = 2$) and small networks ($N = 100$) in general there is not a significant rich-club. This is in agreement with the network growing procedure explained by the PSO model. In fact, the high degree nodes are the first ones to be born in the network and they are expected to connect to around m of the older nodes[20]. Therefore for higher m the rich nodes have higher probability to create a club. Looking at Supplementary Fig. 2-6, it is evident that for networks with $m = [4, 6]$, which are significantly rich-club, the methods using the RA pre-weighting rule do not have any particular decrease in performance, they still provide a very high improvement with respect to the unweighted variant, as for networks with $m = 2$ that do not contain a significant rich-club. We therefore conclude that the RA pre-weighting rule can be adopted regardless of the rich-clubness of a network.

If on one side there are no doubts about the essentialness of the pre-weighting, a discussion is required on the contribution of the equidistant adjustment. In fact, the significant improvement obtained using EA on PSO networks up to 1000 nodes (Fig. 3 and Supplementary Fig. 7) might be due to overfitting to the PSO model, since the angular coordinates are randomly generated by a uniform sampling. Interestingly, Fig. 5 suggests that for PSO networks of size 30000 the EA contribution vanishes. The reason is that with bigger networks the high number of nodes tends to densely and more uniformly cover the angular range hence the non-EA embedding already arranges the nodes in an almost exactly equidistant way. Looking at the community detection application, we noticed that for three out four algorithms (Louvain, Infomap and Walktrap) the EA-based methods obtained in general higher performances than the respective non-EA versions, suggesting that the adjustment of local embedding uncertainty can be effective in real applications. As last, we checked the contribution of the equidistant adjustment on greedy routing. As reported in Supplementary Fig. 20, the equidistant adjustment offered an overall improvement for the greedy routing on PSO networks and a decrease in performance on real networks, although small. To conclude, due to the variable contribution given by the equidistant adjustment, we propose it as a valid alternative to take into consideration, even if it does not represent always the best option. New methods of angular adjustment should be investigated in future studies.

Paper B: A nonuniform popularity-similarity optimization (nPSO) model to efficiently generate realistic complex networks with communities

Alessandro Muscoloni[1] and Carlo Vittorio Cannistraci[1,2,*]

[1] Biomedical Cybernetics Group, Biotechnology Center (BIOTEC), Center for Molecular and Cellular Bioengineering (CMCB), Center for Systems Biology Dresden (CSBD), Department of Physics, Technische Universität Dresden, Tatzberg 47/49, 01307 Dresden, Germany; [2] Brain bio-inspired computing (BBC) lab, IRCCS Centro Neurolesi "Bonino Pulejo", Messina, Italy; * Correspondence should be addressed to: kalokagathos.agon@gmail.com

Published: *New Journal of Physics*, 20, 052002.

Abstract

The investigation of the hidden metric space behind complex network topologies is a fervid topic in current network science and the hyperbolic space is one of the most studied, because it seems associated to the structural organization of many real complex systems. The Popularity-Similarity-Optimization (PSO) model simulates how random geometric graphs grow in the hyperbolic space, generating realistic networks with clustering, small-worldness, scale-freeness and rich-clubness. However, it misses to reproduce an important feature of real complex networks, which is the community organization. The Geometrical-Preferential-Attachment (GPA) model was recently developed in order to confer to the PSO also a soft community structure, which is obtained by forcing different angular regions of the hyperbolic disk to have a variable level of attractiveness. However, the number and size of the communities cannot be explicitly controlled in the GPA, which is a clear limitation for real applications.

Here, we introduce the nonuniform PSO (nPSO) model. Differently from GPA, the nPSO generates synthetic networks in the hyperbolic space where heterogeneous angular node attractiveness is forced by sampling the angular coordinates from a tailored nonuniform probability distribution (for instance a mixture of Gaussians). The nPSO differs from GPA in other three aspects: it allows to explicitly fix the number and size of communities; it allows to tune their mixing property by means of the network temperature; it is efficient to generate networks with high clustering. Several tests on the detectability of the community structure in nPSO synthetic networks and wide investigations on their structural properties confirm that the nPSO is a valid and efficient model to generate realistic complex networks with communities.

B.1 Introduction

In recent years the study of hidden geometrical spaces behind complex network topologies has led to several developments and, currently, the hyperbolic space seems to be one of the most appropriate in order to explain many of the structural features observed in real networks [15], [17]–[20], [30], [32], [51], [65], [68], [74], [81]. In 2012 Papadopoulos et al. [20] introduced the Popularity-Similarity-Optimization (PSO) model in order to describe how random geometric graphs grow in the hyperbolic space optimizing a trade-off between popularity and similarity. In this framework, the popularity of the nodes is represented by the radial coordinate in the hyperbolic disk, whereas the angular coordinates distance is the geometrical counterpart of the similarity between the nodes. Networks generated through the PSO model exhibit strong clustering and a scale-free degree distribution, two among the peculiar properties that usually characterize real-world topologies [11], [37], [38]. However, another important feature commonly observed is the community structure [6], [39], [40], which is lacking in the PSO model. The reason is that the nodes are arranged over the angular coordinate space according to a uniform distribution, therefore, since the connection probability is a decreasing function of the hyperbolic distance, there are not angular regions containing a cluster of spatially close nodes that are more densely connected between each other than with the rest of the network. This issue has been addressed in a following study by Zuev et al. [28], introducing the geometric preferential attachment (GPA). The GPA couples the latent hyperbolic network geometry with preferential attachment of nodes to this geometry in order to generate networks with strong clustering, scale-free degree distribution and a non-trivial community structure [28]. The main assumption of the GPA model and simultaneously the main innovation with respect to the PSO model is that the angular coordinate space is not equally attractive everywhere. Practically, the GPA is characterized by heterogeneous angular attractiveness: regions of different attractiveness are designed according to the rationale that the higher the attractiveness of a region the higher the probability that the nodes are placed in that angular section. Although this general idea can be implemented in several ways, a high-level description of the procedure presented in the study of Zuev et al. [28] is as follows (see Methods for details). For each new node entering in the network, a set of candidate positions is defined (angular coordinate sampled uniformly at random, radial coordinate mathematically fixed) and to every candidate position is assigned a probability depending on the number of nodes that would be “close” to the entering node if it were placed in that position. The probability is also function of a parameter of initial attractiveness, which can be used to tune the heterogeneity of the angular coordinate distribution. However, the GPA model does not allow – at least in the form in which it is currently proposed - to directly control in an *explicit* and *efficient* way the number and size of the communities, a property that instead might be interesting, for example, while proposing a community detection benchmark. Furthermore, the GPA model does not take into account the possibility to vary the network temperature. For this reasons we here introduce a variation of the PSO model, which we call nonuniform PSO (nPSO) model, whose key aspects are the possibility of: a) fixing the number and size of communities; b) tuning their mixing property through the network temperature; c) efficiently producing also highly clustered realistic networks. Finally, although we present the nPSO as a generative model for non-overlapping communities, we will discuss a strategy for taking into account also the presence of overlapping communities.

B.2 Methods

PSO model

The Popularity-Similarity-Optimization (PSO) model [20] is a generative network model recently introduced in order to describe how random geometric graphs grow in the hyperbolic space. In this model the networks evolve optimizing a trade-off between node popularity, abstracted by the radial coordinate, and similarity, represented by the angular coordinate distance, and they exhibit many common structural and dynamical characteristics of real networks.

The model has five input parameters:

- $N > 0$, number of nodes in the network;
- $m > 0$, equal to half of the average node degree;
- $T \geq 0$, network temperature, which controls the network clustering; the network clustering is maximized at $T = 0$, it decreases almost linearly for $T = [0,1)$ and it becomes asymptotically zero if $T > 1$;
- $\beta \in (0, 1]$, popularity fading parameter, or alternatively $\gamma \geq 2$, exponent of the power-law degree distribution, due to the relationship $\gamma = 1 + 1/\beta$;
- $\zeta = \sqrt{-K} > 0$, where K is the curvature of the hyperbolic plane. Since changing ζ rescales the node radial coordinates and this does not affect the topological properties of network [20], in the rest of the article we will consider $K = -1$.

Building a network in the hyperbolic disk requires the following steps:

(1) Initially the network is empty;

(2) At time $i = 1, 2, \dots, N$ a new node i appears with radial coordinate $r_i = 2ln(i)$ and angular coordinate θ_i uniformly sampled in $[0,2\pi]$; all the existing nodes $j < i$ increase their radial coordinates according to $r_j(i) = \beta r_j + (1 - \beta) r_i$ in order to simulate popularity fading;

(3) If $T = 0$, the new node connects to the *m* hyperbolically closest nodes; if $T > 0$, the new node picks a randomly chosen existing node $j < i$ and, given that it is not already connected to it, it connects to it with probability

$$p(i,j) = \frac{1}{1 + \exp\left(\frac{h_{ij} - R_i}{2T}\right)} \tag{1}$$

repeating the procedure until it becomes connected to *m* nodes.

Note that

$$R_i = r_i - 2\ln\left[\frac{2T\left(1 - e^{-(1-\beta)\ln(i)}\right)}{\sin(T\pi)\, m(1-\beta)}\right] \tag{2}$$

is the current radius of the hyperbolic disk, and

$$h_{ij} = arccosh\left(\cosh r_i \cosh r_j - \sinh r_i \sinh r_j \cos\theta_{ij}\right) \tag{3}$$

is the hyperbolic distance between node *i* and node *j,* where

$$\theta_{ij} = \pi - \left|\pi - \left|\theta_i - \theta_j\right|\right| \tag{4}$$

is the angle between these nodes.

(4) The growing process stops when *N* nodes have been introduced.

GPA model

The GPA model is a variation of the original PSO model that couples the latent hyperbolic network geometry with preferential attachment of nodes to this geometry in order to generate networks with strong clustering, scale-free degree distribution and a non-trivial community structure [28].

The procedure to generate a network of *N* nodes is the same described in the previous section for the PSO model, with the main difference that the angular coordinate θ_i of the new node *i* is assigned as follows:

(a) Sample $\varphi_1, \dots, \varphi_i$ in $[0,2\pi]$ uniformly at random. The points (r_i, φ_j) for $j = 1 \dots i$ represent candidate positions for the node.

(b) Define for each candidate position (r_i, φ_j) the attractiveness $A_i(\varphi_j)$ equal to the number of existing nodes that lie within hyperbolic distance r_i from it.

(c) Set the angular coordinate $\theta_i = \varphi_j$ with probability:

$$\Pi_i(\varphi_j) = \frac{A_i(\varphi_j) + \Lambda}{\sum_{k=1}^{i} A_i(\varphi_k) + \Lambda} \tag{5}$$

Where $\Lambda \geq 0$ is a parameter representing the initial attractiveness.

Note that the GPA model has been presented in the related study with only three input parameters, m, β and Λ, with the additional parameters of the PSO model considered in the setting $T = 0$ and $K = -1$.

Nonuniform PSO (nPSO) model

The nonuniform PSO model is a variation of the PSO model introduced in order to confer to the generated networks an adequate community structure, which is lacking in the original model. Since the connection probability is a decreasing function of the hyperbolic distance, a uniform distribution of the nodes over the hyperbolic disk does not create agglomerates of nodes that are concentrated on angular sectors and that are more densely connected between each other than with the rest of the network. A nonuniform distribution, instead, allows to do it by generating heterogeneity in the angular node arrangement. Given the parameters of the PSO model (N, m, T, γ) and a nonuniform probability distribution defined in [0, 2π[, the procedure to generate a network is the same described in the section for the uniform case, with the only difference that the angular coordinates of the nodes are not sampled uniformly but according to the given nonuniform probability distribution.

In particular, without loss of generality, we will concentrate on the mixture of distributions where the components are either Gaussian or Gamma distributions (Figure 1A), which we consider suitable for describing how to build a nonuniform distributed sample of nodes along the angular coordinates of a hyperbolic disk, with communities that emerge in correspondence of the different components. For instance, given a Gaussian mixture distribution the communities will emerge in correspondence of the different Gaussians. In particular, a Gaussian mixture distribution is characterized by the following parameters [113]:

- $C > 0$, which is the number of components, each one representative of a community;
- $\mu_{1 \ldots C} \in [0,2\pi[$, which are the means of the components, representing the central locations of the communities in the angular space;
- $\sigma_{1 \ldots C} > 0$, which are the standard deviations of the components, determining how much the communities are spread in the angular space; a low value leads to isolated communities, a high value makes the adjacent communities to overlap;

- $\rho_{1 \dots C}$ ($\sum_i \rho_i = 1$), which are the mixing proportions of the components, determining the relative sizes of the communities.

Note that, although the means of the components are located in [0, 2π[, the sampling of the angular coordinate θ can fall out of this range. In this case, it has to be shifted within the original range using the modulo operator: $\theta = modulo(\theta, 2\pi)$.

Although the parameters of the Gaussian mixture distribution allow for the investigation of disparate scenarios, as a first case of study (Figure 1B) we focused on the most straightforward setting. For a given number of components *C*, we considered their means equidistantly arranged over the angular space, the same standard deviation and equal mixing proportions:

- $\mu_i = \frac{2\pi}{C} * (i - 1) \quad i = 1 \dots C$
- $\sigma_1 = \sigma_2 = \dots = \sigma_C = \sigma$
- $\rho_1 = \rho_2 = \dots = \rho_C = \frac{1}{C}$

In particular, in our simulations we fixed the standard deviation to 1/6 of the distance between two adjacent means $\left(\sigma = \frac{1}{6} * \frac{2\pi}{C}\right)$, which allowed for a reasonable isolation of the communities independently from their number.

In a second scenario (Figure 1C), we introduced asymmetries in the distribution of the nodes over the circumference by generating communities of different sizes, implemented using diverse mixing proportions for the components. In particular, in our simulations the mixing proportions have been randomly assigned.

As a last scenario (Figure 1D), we considered a mixture of Gaussian and Gamma distributions, which is also characterized by asymmetries due to the presence of the Gamma components. Since this is not a mixture distribution as ordinary as the Gaussian one, in Supplementary Information we will provide the details about how it has been built for our simulations. In all the scenarios, the community memberships are assigned considering for each node the component whose mean is at the lowest angular distance.

Computational implementations of the generative model algorithm

In the PSO model section, at step (3) of the generative procedure, it is presented how the new node establishes connections to *m* of the existing nodes. In particular, if $T > 0$, the new node i picks a randomly chosen existing node $j < i$ and, given that it is not already connected to it, it connects with probability $p(i, j)$, repeating the procedure until it becomes connected to *m* nodes. An example of pseudocode is:

```
targets = [1…i-1]
c = 0
while c < m
        j = random node uniformly sampled from targets
        rand_p = random number in [0,1]
        if p(i,j) > rand_p
                add link from i to j
                remove j from targets
                c = c + 1;
        end
end
```

At the implementation level, the basic solution in MATLAB code would be:

```
targets = 1:(i-1);
c = 0;
while c < m
        idx = randi(length(targets));
        j = targets(idx);
        rand_p = rand(1);
        if p(i,j) > rand_p
                x(i,j) = 1;
                c = c + 1;
                targets(idx) = [];
        end
end
```

where *x* is the adjacency matrix of the network. We will refer to this as implementation 1.

As it will be commented in the Results and Discussion, this implementation has issues of time performance in specific cases. In fact, it is possible to note from Equation (1) that the connection probability $p(i,j)$ decreases both for increasing hyperbolic distance and for decreasing temperature (when $h_{ij} > R_i$, which is true in the majority of the cases as shown in Suppl. Table 1, in particular for increasing network size). Therefore, while generating a network with low temperature and where many nodes are at high hyperbolic distance (for example sampling the angular coordinates from a Gaussian mixture distribution with 4 communities), most of the connection probabilities to the targets will be low. As a consequence, the *if* statement will result *false* in many iterations and the *while* loop will require a relevant computational time before that *m* connections are successfully established.

In order to solve this issue, we note that at each *while*-loop iteration the connection probabilities to the target nodes (excluded the ones already connected) do not always cover the full range

[0, 1]. In particular, at each iteration the maximum of these probabilities $\text{max\_p} = \max_{t \in targets} p(i,t)$ will be usually lower than 1. Since it is known a priori that any random sampling $\text{rand\_p} > \text{max\_p}$ will necessarily bring to a rejection of the connection independently from the target node chosen, the sampling range of the random probability rand_p can be restricted to [0, max_p]. In the critical conditions previously mentioned, where most of the connection probabilities are low, this adjustment can bring to a considerable speedup without biasing the link generation procedure. An example of pseudocode is:

```
targets = [1…i-1]
c = 0
max_p = $\max_{t \in \text{targets}} p(i,t)$
while c < m
        j = random node uniformly sampled from targets
        rand_p = random number in [0,max_p]
        if p(i,j) > rand_p
                add link from i to j
                remove j from targets
                max_p = $\max_{t \in \text{targets}} p(i,t)$
                c = c + 1;
        end
end
```

In case a programming language optimized for vector operations (i.e. MATLAB) is used, since vector operations are faster than loop-based operations, at each iteration *m* attempts of connection to target nodes can be done at once, reducing the number of iterations required to successfully establish *m* connections. Note that, while this adjustment is convenient only at the implementation level when using a programming language optimized for vectorization, the restriction of the probability sampling to the range [0, max_p] is valid in general.

The MATLAB code of the implementation would be:

```
targets = 1:(i-1);
c = 0;
max_p = max(p(i,targets));
while c < m
        if length(targets) > m
                idx = randsample(length(targets),m);
        else
                idx = 1:length(targets);
        end
        rand_p = rand(1,length(idx)) * max_p;
        idx = idx(p(i,targets(idx)) > rand_p);
        if ~isempty(idx)
                if length(idx) > m - c
                        idx = randsample(idx,m - c);
                end
                x(i,targets(idx)) = 1;
                targets(idx) = [];
                max_p = max(p(i,targets));
                c = c + length(idx);
        end
end
```

We will refer to this as implementation 2.

A further variant that we propose is to sample the target nodes according to the theoretical probabilities $p(i,j)$. This solution ensures that at every iteration new connections are successfully established, avoiding rejections and making the procedure faster. An example of pseudocode is:

```
targets = [1…i-1]
for c = 1…m
        j = random node t sampled from targets with probabilities $\frac{\mathrm{p(i,t)}}{\sum_{\mathrm{u}\,\in\,\mathrm{targets}} \mathrm{p(i,u)}}$
        add link from i to j
        remove j from targets
end
```

Given normalized connection probabilities from node i to U targets, $w(i,t) = \frac{\mathrm{p(i,t)}}{\sum_{\mathrm{u}\,\in\,\mathrm{targets}} \mathrm{p(i,u)}}$, the nonuniform sampling can be performed in the following way:

(a) Partition the interval [0,1] in U subintervals $I(t)$ of sizes $w(i,t)$

(b) Generate a random number $r \in [0,1]$

(c) The sampled target is the t such that $r \in I(t)$

The MATLAB code of the implementation would be:

```
targets = 1:(i-1);
idx = datasample(targets, m, 'Replace', false, 'Weights', p(i,targets));
x(i,idx) = 1;
```

We will refer to this as implementation 3.

Note that, as for the previous implementation, the sampling of m targets at once is an adjustment convenient only at the implementation level when using a programming language optimized for vectorization, whereas the idea of sampling according to the theoretical probabilities is valid in general.

The computational complexity of the model using the three implementations is discussed in the next section, whereas their running time as well as the equivalence of the generated synthetic networks is discussed in the Results and Discussion.

Computational complexity

The generative procedure of the nPSO model mainly consists in a loop of *N* iterations, where for each iteration *i* a new node appears and connects to *m* of the existing nodes.

Let's firstly consider the degenerate case in which $m \approx N$. For approximately all the *N* iterations the connections m to establish are more than the existing nodes, therefore the new node i will simply connect to all the previous i-1 nodes, with $O(i)$ operations. The computational complexity is given by:

$$\sum_{i=1}^{N} i = \frac{N \cdot (N+1)}{2} = O(N^2)$$

Let's now consider the more realistic case in which $m \ll N$. For approximately all the *N* iterations the connections *m* to establish are less than the existing nodes, therefore the new node *i* will connect to only *m* of them, and the time-dominant operations to create the links change depending on the temperature (whether zero or positive) and on which of the three implementations is adopted.

For T = 0, the connections are established with the *m* hyperbolically closest nodes, independently from the implementation. For each iteration i, i-1 hyperbolic distances have to be computed in $O(i)$ operations, and then the *m* smallest ones have to be found, which can be obtained building a min-heap in $O(i)$ and retrieving the minimum *m* times in $O(m \log i)$. Considering $\sum_{i=1}^{N} \log i = O(N\log N)$ and $E = mN$, the computational complexity is given by:

$$\sum_{i=1}^{N} (i + m \log i) = \sum_{i=1}^{N} i + m \sum_{i=1}^{N} \log i = O(N^2) + m \cdot O(N\log N) = O(N^2 + E\log N)$$

For T > 0, for each iteration i the m links are instead established according to the connection probabilities $p(i,j)$. We will now analyse the three different implementations.

Using the implementation 1, for each link to create, one target node *t* is uniformly sampled and the connection is established with probability $p(i,t)$, otherwise rejected. Therefore a connection attempt costs constant time $O(1)$. The average connection probability to the targets changes over the iterations *i* and over the *m* links depending on the set of targets. Let's indicate with $\tilde{p}_1$ the average connection probability to the targets over the entire generative procedure using implementation 1. For each iteration i, on average $\frac{m}{\tilde{p}_1}$ connection attempts of cost $O(1)$ are performed and therefore at most $O\left(\frac{m}{\tilde{p}_1}\right)$ operations are required. The computational complexity is given by:

$$\sum_{i=1}^{N} \frac{m}{\tilde{p}_1} = O\left(N\frac{m}{\tilde{p}_1}\right) = O\left(E\frac{1}{\tilde{p}_1}\right)$$

Using the implementation 2, for each link to create, one target node *t* is uniformly sampled and the connection is established with probability $p(i,t)/\text{max\_p}$, otherwise rejected (max_p is the maximum probability between the targets). Computing the maximum costs $O(i)$ and since it has to be updated every time a connection is successfully established, for each iteration i its overall cost is $O(m \cdot i)$. Analogously to implementation 1, on average $\frac{m}{\tilde{p}_2}$ connection attempts of cost $O(1)$ are performed, where $\tilde{p}_2$ is the average connection probability to the targets over the entire generative procedure using implementation 2. The computational complexity is given by:

$$\sum_{i=1}^{N} \left(m \cdot i + \frac{m}{\tilde{p}_2}\right) = O(mN^2) + O\left(N\frac{m}{\tilde{p}_2}\right) = O\left(EN + E\frac{1}{\tilde{p}_2}\right) = O\left(E\left(N + \frac{1}{\tilde{p}_2}\right)\right)$$

Using the implementation 3, for each link to create, one target node *t* is nonuniformly sampled with probabilities $w(i,t) = \frac{\mathrm{p(i,t)}}{\sum_{\mathrm{u} \in \mathrm{targets}} \mathrm{p(i,u)}}$ and the connection is successfully established. The computation of the normalized probabilities and the nonuniform sampling have a cost of $O(i)$, which is performed exactly *m* times. The computational complexity is given by:

$$\sum_{i=1}^{N} m \cdot i = O(mN^2) = O(EN)$$

Note that the factor $\frac{1}{\tilde{p}_1}$ (in implementation 1) is expected to be higher with respect to $\frac{1}{\tilde{p}_2}$ (implementation 2) and they both mainly increase for low temperatures and when the hyperbolic distances are overall high (i.e. low number of communities). Therefore, depending on the model's parameter combination, the factors $\frac{1}{\tilde{p}_1}$ and $\frac{1}{\tilde{p}_2}$ can have a significant impact on the computational time, which will be discussed in Results and Discussion.

B.3 Results and Discussion

The idea behind the nPSO is quite intuitive. The sampling of the angular coordinates from a uniform distribution - which is used by the standard PSO - can be generalized to sampling from any distribution with a desired shape. In particular, a nonuniform distribution would indicate the presence of regions with different levels of node attractiveness. In this study, without loss of generality, we will concentrate on the mixture of distributions where the components can be either Gaussian or Gamma distributions, which we consider suitable for describing how to build a nonuniform distributed sample of nodes along the angular coordinates of a hyperbolic disk, with communities that emerge in correspondence of the different distribution components. However, we want to stress that our nPSO model is general and can be implemented considering any mixture of desired distributions from which to sample the angular coordinates of the nodes.

Although the parameters of the Gaussian and Gamma mixture distributions built on the angular coordinate space allow for the investigation of disparate scenarios, in this work we focused on three straightforward settings, which are illustrated in Figure 1. For a given number of communities C, in the first scenario (Figure 1B), we consider a Gaussian mixture distribution of C components with the means equidistantly arranged over the angular space, the same standard deviation and equal mixing proportions (see Methods for details). In a second scenario (Figure 1C), we introduced asymmetries in the distribution of the nodes over the circumference by generating communities of different sizes, implemented using diverse mixing proportions for the components. As a third and last scenario (Figure 1D), we considered a mixture of Gaussian and Gamma distributions, which is also characterized by asymmetries due to the presence of the Gamma components. In all the scenarios, the community memberships are assigned considering for each node the component whose mean is at the lowest angular distance. Figure 2 shows examples of networks in the hyperbolic space generated using the nPSO model (first scenario, Gaussian mixture distribution with equal proportions is considered) for different values of clustering (temperature, $T = 0.1, 0.5, 0.9$) and community number ($C = 4, 8$), while keeping the other parameters fixed ($N = 100$, $m = 5$, $\gamma = 3$). The related communities are also highlighted using different node colours. The figure indicates below each network also the Normalized Mutual Information (NMI) [98], a measure of performance for evaluating the community detection, computed by comparing the nPSO ground-truth communities and the ones detected by Louvain [44], which is one of the state-of-the-art community detection algorithms [48] (see Supplementary Methods for details). We notice that the communities are perfectly detected both for $C = 4$ and $C = 8$ at low temperature, suggesting that a meaningful community structure is generated by the proposed model. For the same number of communities, if the temperature is increased the performance slightly decreases, because more inter-community links are established in the network, causing as expected a higher rate of wrong assignments by the community detection algorithm.

The next sections will be organized as follows: at first we will prove the equivalence of the three implementations for the generative model algorithm and we will discuss their computational efficiency; later - using the fastest implementation (which is the implementation 3) to generate numerous networks over diverse parameter combinations - we will propose a wide investigation on the detectability of the communities and on the topological properties of the synthetic networks generated by the nPSO.

Equivalence of the three implementations for link generation

Let's consider a node *i* that has to establish a connection with one over U target nodes.

Implementation 1: one target node *t* is chosen uniformly at random and a connection is established with probability $p(i,t)$.

Let's call C_t the event: node *i* connects with target *t*. The probability of the event is:

$$P(C_t) = \frac{1}{U} \cdot p(i,t)$$

Let's call *C* the event: node *i* connects with any of the targets. Taking into account that the event *C* is the union of the events $C_1, C_2, \dots, C_U$ and that these events are mutually exclusive, the probability of the event is:

$$P(C) = P\left(\bigcup_{t \in U} C_t\right) = \sum_{t \in U} P(C_t) = \sum_{t \in U} \frac{1}{U} \cdot p(i,t) = \frac{1}{U} \cdot \sum_{t \in U} p(i,t)$$

In case the connection is rejected, another attempt is iteratively made until node *i* connects with any of the targets. In other words, the procedure is repeated until the event *C* occurs.

Therefore, the probability to eventually recruit the target *t* as a neighbour is given by the conditional probability that node *i* has connected with target *t*, given that event *C* occurred:

$$P(C_t|C) = \frac{P(C_t \cap C)}{P(C)} = \frac{P(C_t)}{P(C)} = \frac{\frac{1}{U} \cdot p(i,t)}{\frac{1}{U} \cdot \sum_{u \in U} p(i,u)} = \frac{p(i,t)}{\sum_{u \in U} p(i,u)}$$

Implementation 2: one target node *t* is chosen uniformly at random and a connection is established with probability $\frac{p(i,t)}{\max\limits_{u \in U} p(i,u)}$.

Following the same procedure as the implementation 1, we obtain:

$$P(C_t) = \frac{1}{U} \cdot \frac{p(i,t)}{\max\limits_{u \in U} p(i,u)} = \frac{1}{U} \cdot \frac{1}{\max\limits_{u \in U} p(i,u)} \cdot p(i,t)$$

$$P(C) = \frac{1}{U} \cdot \sum_{t \in U} \frac{p(i,t)}{\max\limits_{u \in U} p(i,u)} = \frac{1}{U} \cdot \frac{1}{\max\limits_{u \in U} p(i,u)} \cdot \sum_{t \in U} p(i,t)$$

$$P(C_t|C) = \frac{P(C_t)}{P(C)} = \frac{\frac{1}{U} \cdot \frac{1}{\max\limits_{u \in U} p(i,u)} \cdot p(i,t)}{\frac{1}{U} \cdot \frac{1}{\max\limits_{u \in U} p(i,u)} \cdot \sum_{u \in U} p(i,u)} = \frac{p(i,t)}{\sum_{u \in U} p(i,u)}$$

Implementation 3: one target node *t* is chosen non-uniformly at random with probability $\frac{p(i,t)}{\sum_{u \in U} p(i,u)}$ and a connection is established.

Note that in this procedure the connection is never rejected. Therefore we obtain:

$$P(C_t) = \frac{p(i,t)}{\sum_{u \in U} p(i,u)}$$

$$P(C) = 1$$

$$P(C_t|C) = \frac{P(C_t)}{P(C)} = \frac{p(i,t)}{\sum_{u \in U} p(i,u)}$$

Since the probability $P(C_t|C)$ to eventually recruit the target *t* as a neighbour is the same for the three implementations, their equivalence is proven.

However, as a further demonstration that the generative procedure is not biased toward networks with different properties, in Suppl. Tables 2-6 we report for each PSO and nPSO parameter combination some of the main deterministic (in the sense that the measure does not depend by the stochastic generation of null models) topological measures of the networks generated using the three different implementations: clustering coefficient, characteristic path length, assortativity, LCP-correlation and power-law exponent. The results confirm that these structural properties of the model are well preserved and there are not noteworthy changes introduced by the adoption of the algorithmic variants.

nPSO algorithm efficiency in generating networks with high clustering

One of the main drawbacks to use the original algorithmic implementation to establish links adopted by the PSO and GPA models also for the nPSO model, is the lack of efficiency in generating networks with communities characterized by high clustering (low temperature), when $T > 0$. As reported in Figure 3 and Suppl. Figures 1-3, the computational time for generating PSO networks of size *N* = 1000 is in the order of seconds, whereas for nPSO networks with low temperature $T = 0.1$ it might take almost one hour ($C = 8$) or up to several hours ($C = 4$), depending on the number of communities.

The main reason is the following. Assuming $T > 0$, at each time step i of the generative procedure the new node i picks a randomly chosen existing node $j < i$ and, given that it is not already connected to it, it connects with probability $p(i,j)$, repeating the procedure until it becomes connected to m nodes. However, it is possible to note from Equation (1) that the connection probability $p(i,j)$ decreases both for increasing hyperbolic distance and for decreasing temperature (when $h_{ij} > R_i$, which is true in the majority of the cases as shown in Suppl. Table 1, in particular for increasing network size). Therefore, while generating a network with low temperature and where many nodes are at high hyperbolic distance (for instance: a nonuniform PSO model that displays communities presents hyperbolic distances significantly higher than a classical uniform PSO), most of the connection probabilities to the targets will be low. As a consequence, many iterations will be required before that m connections are successfully established. Note that, in the nPSO, the lower the C the higher the distance between adjacent communities, therefore more target nodes will be at high hyperbolic distance, which results in an increased computational time, as pointed out comparing $C = 4$ (Suppl. Figure 2) and $C = 8$ (Suppl. Figure 3). Furthermore, in Figure 3 it can be noticed that the running time increases also for decreasing m. Although this might result counterintuitive because less links need to be generated, the reason is that for decreasing m the radius R_i of the hyperbolic disk (see Equation (1)) decreases, as a consequence also the connection probabilities $p(i,j)$ decreases and therefore more iterations will be required before that m connections are successfully established.

Here we propose two different algorithmic implementations, whose details are provided in the Methods. Figure 3 shows that both the implementations do not present any issue for generating nPSO networks with low temperature. As highlighted in Suppl. Figures 4-5, the fastest is implementation 3, whose key idea is to sample the target nodes according to the theoretical probabilities $p(i,j)$, and it only requires 5 minutes to generate large-size nPSO networks of $N = 10000$, regardless of the temperature. This is indeed expected since it is the only implementation in which the attempts of establishing new connections are never rejected, and its computational complexity is only dependent on the number of nodes and edges, $O(EN)$. On the contrary, the time complexity of the implementations 1 and 2 have a dependency on the average connection probability to the targets during the generative procedure, which is mainly affected by the temperature and by the extent of the hyperbolic distances. Comparing the complexity of the three implementations and taking into account the computational time of the numerical experiments, we can derive that:

$$O(EN) < O\left(E\left(N + \frac{1}{\tilde{p}_2}\right)\right) < O\left(E\frac{1}{\tilde{p}_1}\right)$$

And therefore:

$$O(N) < O\left(N + \frac{1}{\tilde{p}_2}\right) < O\left(\frac{1}{\tilde{p}_1}\right)$$

This result mainly suggests that, in particular in the scenario of low temperature where the time difference is considerably high, the average number of attempts required by implementation 1 to establish one connection has an order of complexity higher than the number of nodes in the network.

Suppl. Figures 6-7 report the time performance for generating GPA networks both with $\Lambda = 0.1$ and $\Lambda = 1$. The advantage of the implementation 3 for low temperature $T = 0.1$ is clearly evident and the computational time difference with implementation 1 becomes more significant for low initial attractiveness $\Lambda = 0.1$. Indeed, in this parameter configuration ($T = 0.1$, $\Lambda = 0.1$) and using implementation 1, the generation of networks with $N = 500$ required several hours and networks with $N = 1000$ were still not generated after one month of running time, whereas implementation 3 required around 1 minute for $N = 1000$. We let notice that a lower initial attractiveness tends to locate new coming nodes in regions where other nodes are already present, generating a lower number of denser regions in the hyperbolic disk. The explanation of the higher computational time with respect to $\Lambda = 1$ is therefore analogous to the one given for lower C in the nPSO model.

Detectability and mixing property of the nPSO communities

The main novelty introduced by the nPSO model with respect to the GPA model is the possibility to generate a tailored community structure at any given temperature different from $T = 0$. Therefore this section of the paper will lead the reader through a wide investigation on the parameter combinations of the nPSO model for which the emerging communities are detectable by a state-of-the-art algorithm.

Synthetic networks have been generated using the nPSO model with parameters $N = [100, 500, 1000]$, $m = [5, 10, 15, 20]$, $T = [0, 0.3, 0.6, 0.9]$, $\gamma = 3$ and angular coordinates sampled according to mixture distributions with components $C = [5, 10, 15, 20]$ for the three different scenarios previously mentioned: Gaussian mixture with equal proportions (nPSO1), Gaussian mixture with random proportions (nPSO2), Gaussian and Gamma mixture with equal proportions (nPSO3). For more details on the parameters of the mixture distributions please refer to the Methods.

For each network the Louvain community detection algorithm [44] has been executed and the communities detected have been compared to the annotated ones computing the Normalized Mutual Information (NMI) [98] (see Supplementary Methods for details). We decided to use the Louvain algorithm because it is a model-free and unsupervised heuristic method for community detection based on modularity optimization [44], therefore its performance is not dependent by any assumption on the generative model, and its results should be robust enough regardless of the generative model used to create the synthetic networks. In addition, the Louvain algorithm has been regarded as one of the most effective algorithms for community detection in previous studies across many real and synthetic datasets [48], [50], [114]. However, the fact that here we tested the community detectability of the nPSO synthetic networks using only the Louvain algorithm is overcome in a second study where we compare the performance of several state-of-the-art algorithms for community detection across different parameters of the nPSO model [115].

The heatmap in Figure 4 reports the mean NMI over 10 repetitions for each parameter combination. The point that firstly captures the attention is the overall higher detectability of the communities in networks of larger size with a lower number of communities (top-right area of the heatmap) in comparison to networks of smaller size with a higher number of communities (bottom-left area of the heatmap). This result suggests that, independently from the kind of mixture distribution (nPSO1, nPSO2 or nPSO3), the ratio between the number of communities (*C*) and the network size (*N*) is a factor that strongly affects the detectability of the nPSO communities. This is indeed expected since, for a fixed network size, the lower the number of communities the higher their separation in the angular space. Since connection probabilities depend on geometrical distances, a higher separation leads to a higher percentage of intra-community links with respect to inter-community links (lower community mixing). Previous studies have already demonstrated that the communities are easier to be detected for a lower *community mixing* [42], [50], defined as the average proportion of links from a node to external communities [42].

The heatmap in Figure 5 reports the mean community mixing over 10 repetitions for the same parameter combinations as in Figure 4. It clearly illustrates that the mixing increases with the number of communities for fixed network size, and it decreases with the network size for fixed number of communities. As expected, the mixing grows also with the temperature, since there is higher probability for a node to establish connections with nodes located far apart from its own community. Furthermore, it increases also with the parameter *m*, in particular for higher *C*, where the nodes have too many links with respect to the community size and they are forced to create edges with external communities.

Suppl. Figure 8 highlights some particular parameter combinations from the heatmap in Figure 4 and this helps to discuss some counterintuitive scenarios due to the parameter combinations. Suppl. Figure 8A focuses on small networks (*N* = 100) with a low number of communities (*C* = 5) for the first scenario (nPSO1), and it shows that for increasing temperature the NMI decreases for *m* = 5 whereas tends to increase for *m* = 20. This is reasonable because, when each node creates few connections (*m* = 5), directing them towards external communities (higher temperature) makes the community structure less detectable (lower NMI). Instead, when too many links are generated (*m* = 20), a high temperature avoids that most of the inter-communities links are directed to adjacent communities and helps to make more distinct the community boundaries.

Suppl. Figure 8B reports similar results but from a different perspective. It shows that for increasing *m* the NMI decreases for *T* = 0 whereas tends to increase for *T* = 0.9. In fact, at *T* = 0 most of the links are internal to the community and increasing *m* will only increase the links external to the community, being its size small. On the contrary, at *T* = 0.9 many links are also directed to other communities and increasing *m* will help to have enough internal links to make the communities better detectable. These patterns highlighted on small networks (*N* = 100) with few communities (*C* = 5) are mainly preserved also for higher number of communities, although the overall detectability decreases, as already discussed.

For networks of larger size (*N* = [500, 1000]) with few communities (*C* = 5) the detectability is generally high and tends to be better for middle temperatures. For increasing *C* the NMI overall decreases, and the highest detectability occurs at low *m* for middle temperatures (see Suppl. Figure 8C). The reason is that, due to the larger number of nodes and communities, at *T* = 0 there is higher probability for the nodes to link to adjacent communities, making less distinct the boundaries, while at *T* = 0.9 there is higher probability to lose the preferentiality of connection to nodes of the same community. Middle temperatures guarantee a good proportion between links internal to the community and links directed to all the other communities (not only the adjacent ones).

Suppl. Figure 8D focuses on networks of *N* = 1000 nodes with a high number of communities (*C* = 20) for the first scenario (nPSO1). It shows that for increasing *m* the NMI decreases for *T* = 0 whereas it is almost not affected for *T* = 0.9. As discussed for the smaller networks, at *T* = 0 most of the links are internal to the community and increasing *m* will mainly increase the links external to the community, making it less detectable. At $T = 0.9$, differently from what is shown in Suppl. Figure 8B, the NMI remains almost constant and does not increase with m, probably due to the fact that the communities are bigger and therefore more internal links are required to make them better detectable.

Finally, although there are some minor variabilities between the different scenarios (nPSO1, nPSO2 and nPSO3), the patterns discussed are mostly consistent over all the nPSO model parameter combinations.

Topological properties of the nPSO networks

After having proposed a wide investigation on the detectability of the communities generated by the nPSO, in this section we are going to highlight to which extent the community organization affects the main structural properties of the synthetic networks. For the same parameter combinations of the nPSO model as in Figure 4, and considering also synthetic networks generated using the (conventional) PSO model with the same parameters N, m, T and γ, we computed several topological measures: clustering coefficient, characteristic path length, assortativity, LCP-correlation, structural consistency, power-law exponent, modularity, small-worldness and rich-clubness. The related heatmaps are reported in Figures 6-10 and Suppl. Figures 9-12 and will be now discussed.

Figure 6 shows the clustering coefficient, which offers an average evaluation of the cross-interaction density between the first neighbours of each node in the network [4]. The clustering coefficient strongly decreases for increasing temperature, since there is higher probability to establish connections between nodes that are far apart from each other, and therefore it is less likely to close triangles in a node neighbourhood. Increasing m tends to increase the clustering coefficient, in fact with a higher number of links it is also more likely to close local triangles, and this is more evident on small networks, where there are less target nodes to connect. The type of angular coordinate distribution (nPSO1, nPSO2 or nPSO3) and the number of communities (C) does not have a remarkable effect for most of the parameter combinations. The few cases in which there is higher variability are for $N = [500, 1000]$ and $T = 0.3$ where, with the increase of the number of communities, the clustering coefficient increases, becoming closer and closer to the one of the PSO model.

Figure 7 shows the characteristic path length, which describes the average of the shortest path lengths between all the pairs of vertices [4]. The measure decreases for increasing temperature, since there is higher probability to establish connections between nodes that are far apart from each other, acting as bridges between different, and often far apart, regions of the network. This decrease of the characteristic path length is attenuated when there are many edges with respect to the network size (higher *m* and lower *N*), because the bridges naturally emerge due to the high network density. Increasing *m*, indeed, leads in general to a decrease of the characteristic path length. The type of angular coordinate distribution (nPSO1, nPSO2 or nPSO3) and the number of communities (*C*) does not have any remarkable effect.

Figure 8 reports the assortativity, which indicates the tendency of the networks to connect nodes with similar degree [116]. Positive values suggest an assortative behavior and negative values a disassortative mixing. The results highlight that there are no parameter combinations for which the networks are strongly assortative, whereas disassortativity is detected for networks of small size at low temperature. In the generative procedure of the PSO and nPSO models, the oldest nodes have the highest node degree and, being at the center of the hyperbolic disk, at low temperature they tend to be the connection targets of new coming nodes with lower degree, leading to a disassortative mixing. At higher temperature this disassortativity gets weaker and there is more balance between the connections established from a new node to popular and less popular nodes, resulting in an increase of the measure. It can be noticed also a decrease at $N = 100$ and $T = 0$ for increasing *m*, since more connections are created between the older higher-degree nodes and the younger lower-degree nodes. The type of angular coordinate distribution (nPSO1, nPSO2 or nPSO3) and the number of communities (*C*) leads only to minor variabilities for low temperature with respect to the PSO.

Figure 9 shows the LCP-correlation, the Pearson correlation between the number of common neighbours (which create a local community) and the number of local community links (connections among common neighbours) that are computed for each link of the network [54]. The LCP-correlation measures whether the network follows a local-community-paradigm (LCP) organization [54]–[56] and therefore whether the network is organized in local communities (one for each link) where the number of interactions between the common neighbours is a function that increases with the number of common neighbours in the local community. Complex adaptive networks with weak-links that make local processing and global delivery generally follow the LCP organization (the LCP-correlation is generally ≥ 0.7), whereas the networks that do not follow the LCP organization (LCP-correlation ≤ 0.3) present strong-links, they are not clustered and they are suitable for storage or mere delivery of energy or information. It is very rare to find networks that have a LCP-correlation between 0.3 and 0.7. For this reason we expect to find that all the nPSO networks are organized according to the LCP, however the level of LCP-correlation might change between 0.7 and 1. Indeed, as expected, the LCP-correlation obtains high values over all the parameter combinations of the network. Since connection probabilities depend on geometrical distances, for a given link of the network it is likely that the adjacent nodes are close in the hyperbolic space, therefore it is likely that their common neighbours (if any) are close, and as a consequence it is also likely that these common neighbours have connections among them that increase with the number of common neighbours. Explained in a simpler form: the smaller the hyperbolic distance between two linked nodes, the more common neighbours exist between them (since the geometrical space that separates the two linked points is smaller, the two linked points share more adjacent nodes, which are in fact common neighbours), as a consequence the smaller geometrical space will generate also more connections between these common neighbours. This mechanism obviously is corrupted for increasing temperature, since the connection probabilities have a weaker dependency on the geometrical distances, and therefore the LCP-correlation decreases for high temperatures. In particular, this temperature-dependent LCP-correlation decrease is remarkable for lower m and higher N, since there are less links to establish and more possible connection targets, which reduces the probability to create both common neighbours and local community links (links between the common neighbours). The type of angular coordinate distribution (nPSO1, nPSO2 or nPSO3) and the number of communities (C) leads only to minor variabilities for $T > 0$.

Figure 10 reports the structural consistency, which quantifies the link predictability of the network, characterizing the inherent difficulty to predict the missing or non-observed links regardless of the specific algorithm used for the prediction [58]. The structural consistency strongly decreases for increasing temperature, in particular from $T = 0$ to $T > 0$. In fact, at $T = 0$ the links are regularly established with the closest target nodes, which makes the structure highly consistent and easier to predict. Furthermore, creating a higher number of connections according to this regular pattern ($T = 0$ and higher m) strengthens even more the consistency of the structure. The link predictability becomes lower for increasing network size, since there are potentially more missing or non-observed links to predict. The type of angular coordinate distribution (nPSO1, nPSO2 or nPSO3) and the number of communities (C) leads only to minor variabilities.

Suppl. Figure 9 shows the exponent γ of the power-law degree distribution, fitted using the procedure described by Clauset et al. [91], in order to test whether the value provided in input to the PSO and nPSO models is indeed reproduced. The results highlight that all the fitted values are very close to the desired exponent $\gamma = 3$. The variability might be either due to the difficulty of the model to reproduce perfectly the input value or due to some defects in the fitting procedure. The diverse community organization does not introduce a remarkable bias in the degree distribution.

Suppl. Figure 10 reports the modularity, indicating the extent to which the network can be partitioned in segregated modules that tend to interact densely within themselves but sparsely between each other [39]. We let notice that in nPSO networks the modularity is inversely related to the community mixing, since the lower the community mixing the more the network can be partitioned in distinct modules. Indeed, the Pearson correlation between the community mixing and the modularity over all the parameter combinations of the nPSO model is -0.91. The main patterns observed on nPSO networks for the community mixing are therefore valid, in an inverse way, also for the modularity. For PSO networks the modularity is generally lower, with an exception for larger networks, low m and low temperature, probably due to the fact that many small modules naturally emerge since for low temperatures the clustering is very high. The type of angular coordinate distribution (nPSO1, nPSO2 or nPSO3) leads only to minor variabilities, although we let notice that such variabilities might be even due to the randomness in the modularity evaluation procedure.

Suppl. Figure 11 shows the measure of small-worldness ω, which indicates whether a network exhibits a small-world organization, characterized by a clustering coefficient (CL) as high as in an equivalent lattice network (CL_{latt}) and a characteristic path length (L) as low as in an equivalent random network (L_{rand}) [4], [117]: $\omega = \frac{L_{rand}}{L} - \frac{CL}{CL_{latt}}$. The measure ω is expected to be close to 0 in small-world networks ($L \approx L_{rand}$ and $CL \approx CL_{latt}$), higher than 0 for random networks ($L \approx L_{rand}$ and $CL < CL_{latt}$) and lower than 0 for lattice networks ($L > L_{rand}$ and $CL \approx CL_{latt}$). The parameter combinations closest to small-world networks are at $N = 100$, $m = [15, 20]$ and $T = 0$. Indeed, these are the synthetic networks characterized by the highest clustering coefficient and the lowest characteristic path length. The measure ω increases for increasing temperature, since the clustering coefficient strongly decreases and the characteristic path length slightly decreases, with a transition from structural properties of a regular network to the ones of a random network. At $T = 0$, the measure ω increases for increasing m, since the clustering coefficient is constantly high and the characteristic path length decreases. This is not always valid at $T > 0$, where sometimes the increase in clustering balances the decrease of the characteristic path length. The type of angular coordinate distribution (nPSO1, nPSO2 or nPSO3) and the number of communities (C) leads only to minor variabilities.

Suppl. Figure 12 reports the p-value of the statistical test for rich-clubness, which indicates whether the network presents a significant rich-club organization with respect to the

Cannistraci-Muscoloni null-model [110]. It clearly emerges that for almost all the parameter combinations the synthetic networks are significantly (p-value ≤ 0.05) characterized by rich-clubness. This is indeed in agreement with the network growing procedure explained by the PSO and nPSO models. In fact, the high degree nodes are the first ones to be born in the network and they connect to m of the nodes already present [20], therefore every network has at least a fully connected subgraph composed by the m+1 oldest high-degree nodes. The only p-values that are not significant are borderline and detected for small ($N = 100$) and sparse ($m = 5$) networks, since there are less nodes and links to build the rich-club, and only at higher temperature, where the connection probabilities have a weaker dependency from geometrical distances and therefore the rich and popular nodes decrease their attractiveness for new connections. The type of angular coordinate distribution (nPSO1, nPSO2 or nPSO3) and the number of communities (C) leads only to minor variabilities among the borderline cases.

All these topological measures have been evaluated using the MATLAB code released at: https://github.com/biomedical-cybernetics/topological_measures_wide_analysis [118].

B.4 Conclusion

Recent studies presented the hyperbolic disk as an adequate space to describe the latent geometry of real complex networks and the PSO model was introduced to generate random geometric graphs in the hyperbolic space, reproducing strong clustering and a scale-free degree distribution [20]. Coupling the hyperbolic space with the preferential attachment of nodes to this space, the GPA model confers to the networks also a community structure, introducing the idea that different angular regions of the hyperbolic disk can have a variable level of attractiveness [28]. However, the GPA model does not allow to indicate in input a desired number of communities, neither to control their size and the mixing between them, which is a clear limitation for real applications. For this reason, we here introduced the nonuniform PSO (nPSO) model, which allows to explicitly fix the number of communities and their size by means of a tailored probability distribution on the angular coordinates, and to tune the mixing property through the network temperature.

We performed extensive tests on the detectability of the nPSO communities, considering also more complicated settings with asymmetric angular coordinate distributions over the angular space. We highlighted that, for most of the parameter combinations representing realistic scenarios, the community organization can be spotted by the state-of-the-art algorithm Louvain. The main factor that reduces the detectability is the ratio between the number of communities and the network size, in particular community detection in nPSO networks reduces significantly for small size networks that present many communities. These results suggest that realistic community structure is properly reproduced by the model and the nPSO might be employed in future studies as a benchmark for testing community detection algorithms. On this regard, we propose a second study that discusses how to leverage the nPSO model to test and compare the performance of different algorithms for community detection and also link prediction [115].

We evaluated and compared several topological measures of the synthetic networks generated using the PSO and nPSO models, and from this wide investigation two important results emerge. First, the parameters of the model allow to reproduce a great variety of the structural properties observed in real-world complex networks, and the heatmaps provided in this study can be used as a reference for the choice of parameters while generating networks with desired characteristics. Second, the diverse community organization has only a minor impact for most of the main topological measures. This suggests, for example, that the temperature of a real network can be inferred from the clustering coefficient regardless of the community structure. From the algorithmic point of view, since the original procedure to establish links adopted by the PSO and GPA models is computationally expensive for generating networks with communities and high clustering, we proposed other two different variants. We demonstrated that the three implementations generate equivalent topologies and the fastest of them (implementation 3) significantly reduces the computational time, with a complexity of $O(EN)$ independently from the communities and the clustering.

Although in this work we present the nPSO as a generative model for realistic networks with non-overlapping communities, its current implementation would be able also to generate networks with overlapping communities, for instance by increasing the standard deviations of the components in the Gaussian mixture distribution. However, a specific rule according to which the nodes are assigned to one or more communities needs to be designed, depending both on the geometrical positions of the nodes (angular and radial coordinates) and on the mixture distribution parameters. This extension of the nPSO model will be investigated in future studies.

To conclude, we propose the nPSO model as a valid framework able to efficiently generate realistic networks with a fixed number of communities according to a nonuniform node-angular probability distribution. The nPSO might be adopted, among the many possibilities, as a null model for the hyperbolic embedding of networks with community structure, or as a benchmark for testing community detection and link prediction algorithms, as we illustrate and discuss in a second study dedicated to this topic [115].

Code availability

The MATLAB code for generating synthetic networks using the nPSO model is publicly available at the GitHub repository: https://github.com/biomedical-cybernetics/nPSO_model

Hardware and software

MATLAB code has been used for all the simulations, carried out partly on a workstation under Windows 8.1 Pro with 512 GB of RAM and 2 Intel(R) Xenon(R) CPU E5-2687W v3 processors with 3.10 GHz, and partly in the ZIH-Cluster Taurus of the TU Dresden.

Funding

Work in the CVC laboratory was supported by the independent research group leader starting grant of the Technische Universität Dresden. AM was partially supported by the funding provided by the Free State of Saxony in accordance with the Saxon Scholarship Program Regulation, awarded by the Studentenwerk Dresden based on the recommendation of the board of the Graduate Academy of TU Dresden.

Acknowledgements

We thank Alexander Mestiashvili and the BIOTEC System Administrators for their IT support, Claudia Matthes for the administrative assistance and the Centre for Information Services and High Performance Computing (ZIH) of the TUD.

Author contributions

CVC invented the nonuniform PSO model and designed the numerical experiments. AM implemented the code and performed the computational analysis. Both the authors analysed and interpreted the results. AM and CVC built the demonstration of the equivalence of the three implementations for link generation and AM formalized it. AM performed the analysis for the computational complexity and CVC checked it. AM wrote the draft of the article according to CVC suggestions and CVC corrected and improved it to arrive to the final draft. CVC designed the figures and AM realized them. AM designed and realized the heatmap tables. CVC planned, directed and supervised the study.

Competing interests

The authors declare no competing financial interests.

B.5 Figures

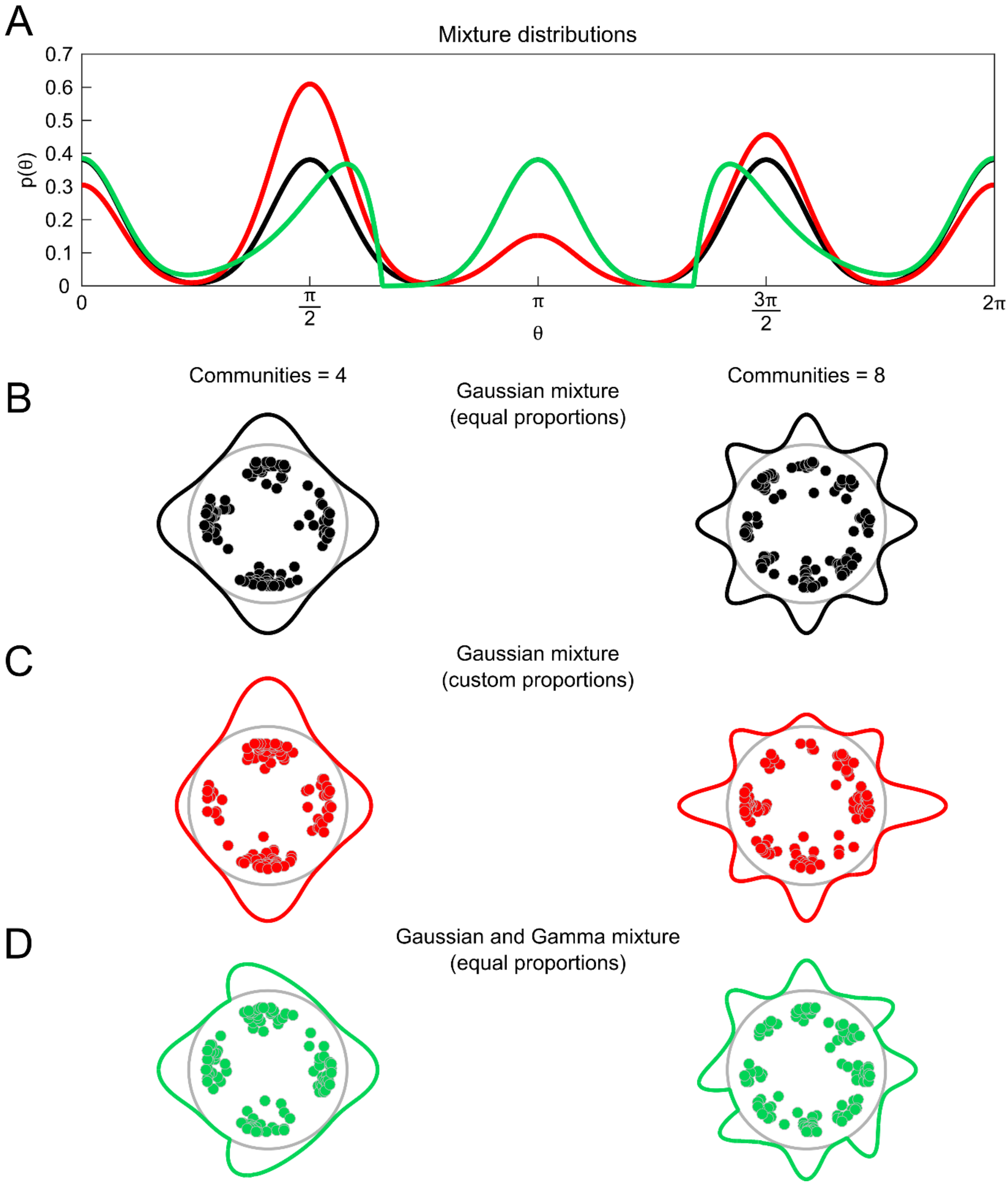

A
Mixture distributions
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
p(θ)
0
π/2
π
3π/2
2π
θ
B
Communities = 4
Gaussian mixture
(equal proportions)
Communities = 8
C
Gaussian mixture
(custom proportions)
D
Gaussian and Gamma mixture
(equal proportions)

Figure 1. Nonuniform distributions of angular coordinates.

(A) Plot of three different kinds of nonuniform distributions used for sampling the angular coordinates of the nodes: Gaussian mixture with equal proportions (nPSO1, black line), Gaussian mixture with custom proportions (nPSO2, red line), Gaussian and Gamma mixture with equal proportions (nPSO3, green line). The mixture distributions have 4 components, placing the mean of the components equidistantly over the angular space. More details on the parameters of the mixture distributions are in the Methods. **(B-C-D)** The same three kinds of mixture distributions in (A) are represented along the angular space of the hyperbolic disk, using 4 and 8 components. For each of them, a synthetic network has been generated using the nPSO model with parameters $N = 100$, $m = 2$, $T = 0$, $\gamma = 3$ and angular coordinates sampled according to the mixture distribution. The coordinates of the nodes are represented in the hyperbolic disk in order to provide an example of the emerging community structure.

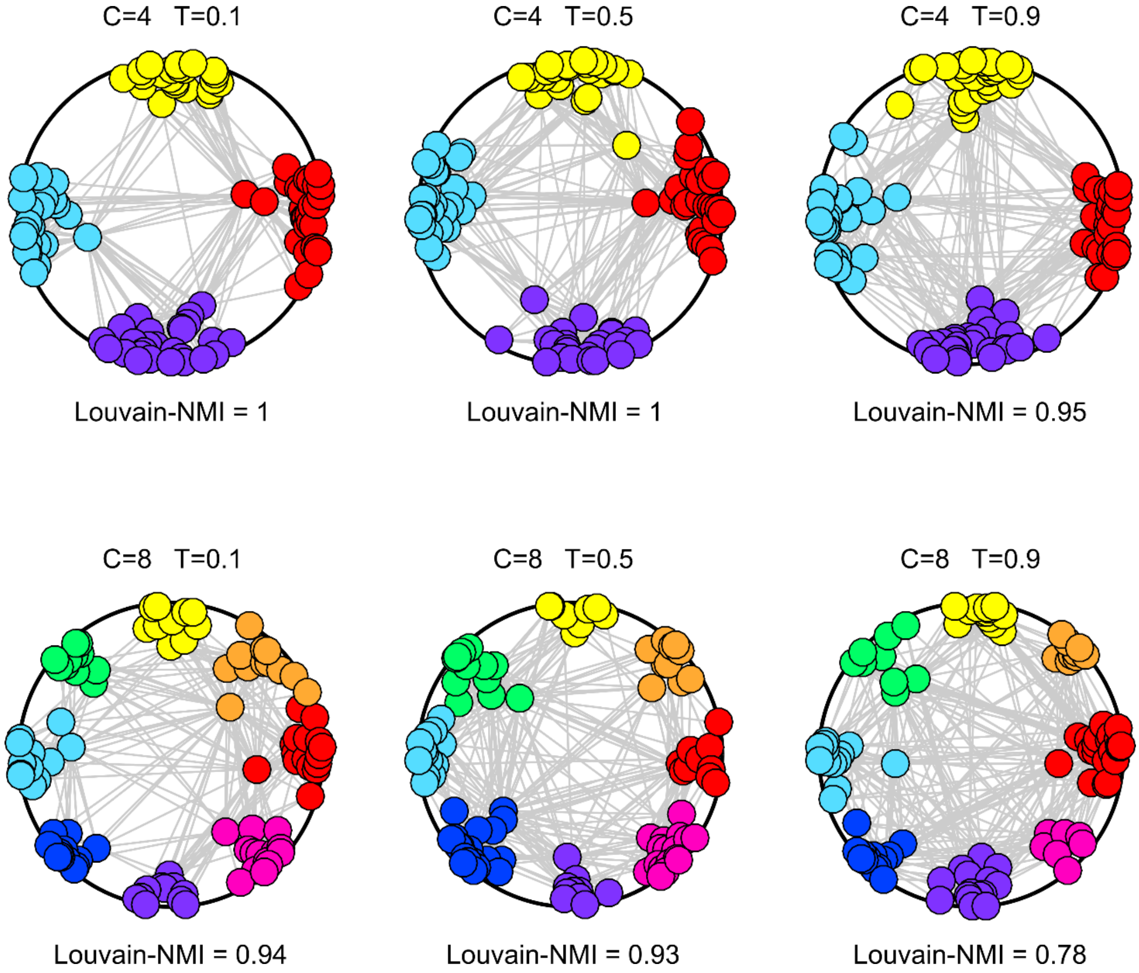


Figure 2. Communities generated using the nPSO model.

Synthetic networks have been generated using the nPSO model with parameters $N = 100$, $m = 5$, $T = [0.1, 0.5, 0.9]$, $\gamma = 3$ and angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components $C = [4, 8]$ (communities). For each combination of parameters, 10 networks have been generated. For each network the Louvain community detection method has been executed and the communities detected have been compared to the annotated ones computing the Normalized Mutual Information (NMI). The plots show for each parameter combination a representation in the hyperbolic space of the network that obtained the highest NMI, whose value is reported. The nodes are coloured according to the communities as generated by the nPSO model.

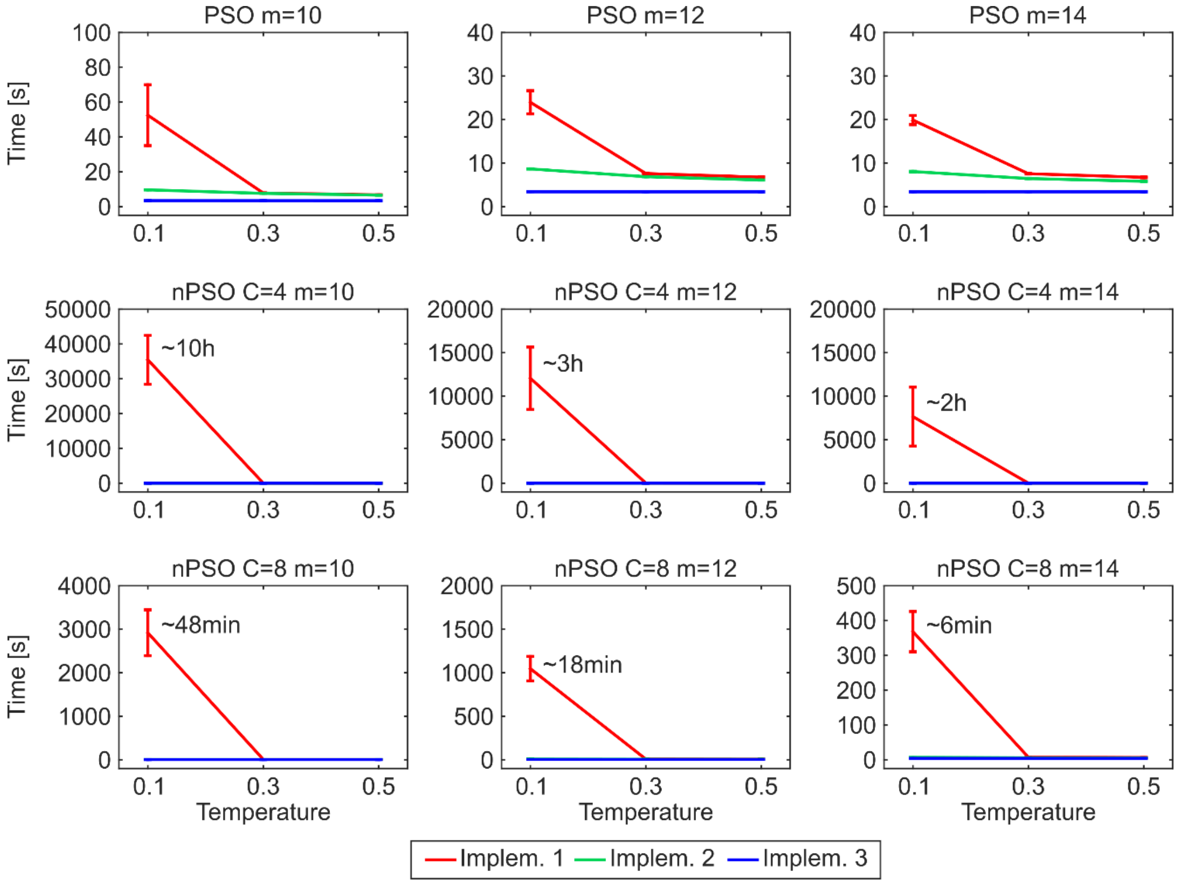
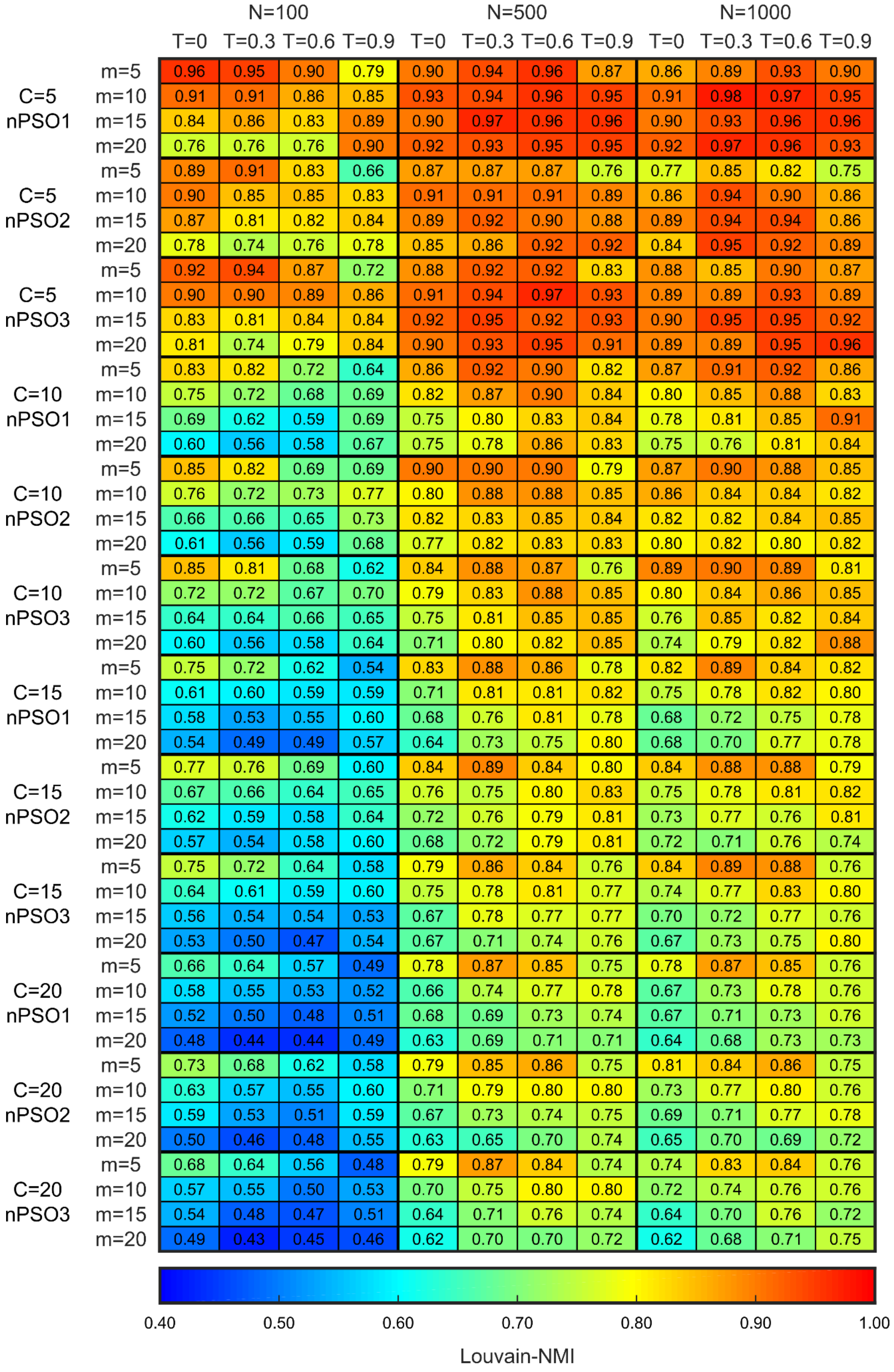


Figure 3. Time performance for generating PSO and nPSO networks.
Synthetic networks have been generated using the PSO and nPSO models with parameters $N = 1000$, $m = [10, 12, 14]$, $T = [0.1, 0.3, 0.5]$, $\gamma = 3$ and, for the nPSO model, angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components $C = [4, 8]$. For each combination of parameters, 10 networks have been generated using the 3 different implementations. The plots report for each parameter combination the mean computational time and standard error over the random repetitions. A more detailed comparison of implementations 2 and 3 and additional plots for different network sizes are provided in Supplementary Information.



		N=100				N=500				N=1000			
		T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9
C=5 nPSO1	m=5	0.96	0.95	0.90	0.79	0.90	0.94	0.96	0.87	0.86	0.89	0.93	0.90
	m=10	0.91	0.91	0.86	0.85	0.93	0.94	0.96	0.95	0.91	0.98	0.97	0.95
	m=15	0.84	0.86	0.83	0.89	0.90	0.97	0.96	0.96	0.90	0.93	0.96	0.96
	m=20	0.76	0.76	0.76	0.90	0.92	0.93	0.95	0.95	0.92	0.97	0.96	0.93
C=5 nPSO2	m=5	0.89	0.91	0.83	0.66	0.87	0.87	0.87	0.76	0.77	0.85	0.82	0.75
	m=10	0.90	0.85	0.85	0.83	0.91	0.91	0.91	0.89	0.86	0.94	0.90	0.86
	m=15	0.87	0.81	0.82	0.84	0.89	0.92	0.90	0.88	0.89	0.94	0.94	0.86
	m=20	0.78	0.74	0.76	0.78	0.85	0.86	0.92	0.92	0.84	0.95	0.92	0.89
C=5 nPSO3	m=5	0.92	0.94	0.87	0.72	0.88	0.92	0.92	0.83	0.88	0.85	0.90	0.87
	m=10	0.90	0.90	0.89	0.86	0.91	0.94	0.97	0.93	0.89	0.89	0.93	0.89
	m=15	0.83	0.81	0.84	0.84	0.92	0.95	0.92	0.93	0.90	0.95	0.95	0.92
	m=20	0.81	0.74	0.79	0.84	0.90	0.93	0.95	0.91	0.89	0.89	0.95	0.96
C=10 nPSO1	m=5	0.83	0.82	0.72	0.64	0.86	0.92	0.90	0.82	0.87	0.91	0.92	0.86
	m=10	0.75	0.72	0.68	0.69	0.82	0.87	0.90	0.84	0.80	0.85	0.88	0.83
	m=15	0.69	0.62	0.59	0.69	0.75	0.80	0.83	0.84	0.78	0.81	0.85	0.91
	m=20	0.60	0.56	0.58	0.67	0.75	0.78	0.86	0.83	0.75	0.76	0.81	0.84
C=10 nPSO2	m=5	0.85	0.82	0.69	0.69	0.90	0.90	0.90	0.79	0.87	0.90	0.88	0.85
	m=10	0.76	0.72	0.73	0.77	0.80	0.88	0.88	0.85	0.86	0.84	0.84	0.82
	m=15	0.66	0.66	0.65	0.73	0.82	0.83	0.85	0.84	0.82	0.82	0.84	0.85
	m=20	0.61	0.56	0.59	0.68	0.77	0.82	0.83	0.83	0.80	0.82	0.80	0.82
C=10 nPSO3	m=5	0.85	0.81	0.68	0.62	0.84	0.88	0.87	0.76	0.89	0.90	0.89	0.81
	m=10	0.72	0.72	0.67	0.70	0.79	0.83	0.88	0.85	0.80	0.84	0.86	0.85
	m=15	0.64	0.64	0.66	0.65	0.75	0.81	0.85	0.85	0.76	0.85	0.82	0.84
	m=20	0.60	0.56	0.58	0.64	0.71	0.80	0.82	0.85	0.74	0.79	0.82	0.88
C=15 nPSO1	m=5	0.75	0.72	0.62	0.54	0.83	0.88	0.86	0.78	0.82	0.89	0.84	0.82
	m=10	0.61	0.60	0.59	0.59	0.71	0.81	0.81	0.82	0.75	0.78	0.82	0.80
	m=15	0.58	0.53	0.55	0.60	0.68	0.76	0.81	0.78	0.68	0.72	0.75	0.78
	m=20	0.54	0.49	0.49	0.57	0.64	0.73	0.75	0.80	0.68	0.70	0.77	0.78
C=15 nPSO2	m=5	0.77	0.76	0.69	0.60	0.84	0.89	0.84	0.80	0.84	0.88	0.88	0.79
	m=10	0.67	0.66	0.64	0.65	0.76	0.75	0.80	0.83	0.75	0.78	0.81	0.82
	m=15	0.62	0.59	0.58	0.64	0.72	0.76	0.79	0.81	0.73	0.77	0.76	0.81
	m=20	0.57	0.54	0.58	0.60	0.68	0.72	0.79	0.81	0.72	0.71	0.76	0.74
C=15 nPSO3	m=5	0.75	0.72	0.64	0.58	0.79	0.86	0.84	0.76	0.84	0.89	0.88	0.76
	m=10	0.64	0.61	0.59	0.60	0.75	0.78	0.81	0.77	0.74	0.77	0.83	0.80
	m=15	0.56	0.54	0.54	0.53	0.67	0.78	0.77	0.77	0.70	0.72	0.77	0.76
	m=20	0.53	0.50	0.47	0.54	0.67	0.71	0.74	0.76	0.67	0.73	0.75	0.80
C=20 nPSO1	m=5	0.66	0.64	0.57	0.49	0.78	0.87	0.85	0.75	0.78	0.87	0.85	0.76
	m=10	0.58	0.55	0.53	0.52	0.66	0.74	0.77	0.78	0.67	0.73	0.78	0.76
	m=15	0.52	0.50	0.48	0.51	0.68	0.69	0.73	0.74	0.67	0.71	0.73	0.76
	m=20	0.48	0.44	0.44	0.49	0.63	0.69	0.71	0.71	0.64	0.68	0.73	0.73
C=20 nPSO2	m=5	0.73	0.68	0.62	0.58	0.79	0.85	0.86	0.75	0.81	0.84	0.86	0.75
	m=10	0.63	0.57	0.55	0.60	0.71	0.79	0.80	0.80	0.73	0.77	0.80	0.76
	m=15	0.59	0.53	0.51	0.59	0.67	0.73	0.74	0.75	0.69	0.71	0.77	0.78
	m=20	0.50	0.46	0.48	0.55	0.63	0.65	0.70	0.74	0.65	0.70	0.69	0.72
C=20 nPSO3	m=5	0.68	0.64	0.56	0.48	0.79	0.87	0.84	0.74	0.74	0.83	0.84	0.76
	m=10	0.57	0.55	0.50	0.53	0.70	0.75	0.80	0.80	0.72	0.74	0.76	0.76
	m=15	0.54	0.48	0.47	0.51	0.64	0.71	0.76	0.74	0.64	0.70	0.76	0.72
	m=20	0.49	0.43	0.45	0.46	0.62	0.70	0.70	0.72	0.62	0.68	0.71	0.75

Figure 4. Detectability of the nPSO communities.

Synthetic networks have been generated using the nPSO model with parameters $N = [100, 500, 1000]$, $m = [5, 10, 15, 20]$, $T = [0, 0.3, 0.6, 0.9]$, $\gamma = 3$ and angular coordinates sampled according to mixture distributions of three different kinds with components $C = [5, 10, 15, 20]$: Gaussian mixture with equal proportions (nPSO1), Gaussian mixture with random proportions (nPSO2), Gaussian and Gamma mixture with equal proportions (nPSO3). For more details on the parameters of the mixture distributions please refer to the Methods. For each combination of parameters, 10 networks have been generated. For each network the Louvain community detection method has been executed and the communities detected have been compared to the annotated ones computing the Normalized Mutual Information (NMI). The heatmap reports for each parameter combination the mean NMI, coloured according to a blue-to-red colormap in the range [0.4, 1].

		N=100				N=500				N=1000			
		T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9
C=5 nPSO1	m=5	0.09	0.18	0.29	0.36	0.02	0.04	0.11	0.22	0.01	0.02	0.08	0.18
	m=10	0.19	0.34	0.44	0.47	0.04	0.08	0.17	0.27	0.02	0.04	0.11	0.22
	m=15	0.31	0.46	0.53	0.52	0.05	0.11	0.22	0.31	0.03	0.06	0.14	0.25
	m=20	0.41	0.55	0.59	0.58	0.07	0.14	0.26	0.33	0.04	0.07	0.17	0.27
C=5 nPSO2	m=5	0.10	0.17	0.27	0.32	0.02	0.04	0.11	0.20	0.01	0.02	0.07	0.16
	m=10	0.18	0.30	0.39	0.40	0.04	0.07	0.15	0.22	0.02	0.04	0.11	0.19
	m=15	0.27	0.41	0.47	0.46	0.05	0.10	0.19	0.25	0.03	0.06	0.13	0.23
	m=20	0.36	0.48	0.48	0.48	0.08	0.14	0.23	0.30	0.04	0.07	0.15	0.23
C=5 nPSO3	m=5	0.11	0.18	0.30	0.37	0.02	0.04	0.12	0.23	0.01	0.02	0.08	0.19
	m=10	0.20	0.34	0.44	0.47	0.04	0.08	0.18	0.28	0.02	0.04	0.12	0.23
	m=15	0.32	0.47	0.53	0.54	0.06	0.12	0.22	0.31	0.03	0.06	0.15	0.25
	m=20	0.41	0.55	0.59	0.59	0.08	0.15	0.26	0.34	0.04	0.08	0.18	0.28
C=10 nPSO1	m=5	0.21	0.35	0.48	0.53	0.04	0.07	0.19	0.32	0.02	0.04	0.13	0.27
	m=10	0.41	0.57	0.63	0.64	0.07	0.14	0.28	0.39	0.04	0.07	0.19	0.32
	m=15	0.56	0.69	0.71	0.70	0.11	0.21	0.35	0.43	0.06	0.11	0.24	0.36
	m=20	0.67	0.76	0.77	0.75	0.16	0.27	0.41	0.48	0.07	0.14	0.28	0.39
C=10 nPSO2	m=5	0.18	0.32	0.44	0.47	0.04	0.07	0.17	0.29	0.02	0.04	0.12	0.24
	m=10	0.37	0.51	0.55	0.56	0.07	0.13	0.26	0.35	0.04	0.08	0.17	0.29
	m=15	0.48	0.63	0.65	0.62	0.11	0.20	0.32	0.39	0.05	0.11	0.21	0.32
	m=20	0.57	0.72	0.72	0.69	0.14	0.25	0.37	0.42	0.08	0.13	0.25	0.34
C=10 nPSO3	m=5	0.22	0.35	0.48	0.54	0.04	0.08	0.20	0.34	0.02	0.04	0.13	0.28
	m=10	0.42	0.58	0.64	0.65	0.08	0.15	0.29	0.40	0.04	0.08	0.19	0.33
	m=15	0.58	0.69	0.72	0.71	0.13	0.22	0.37	0.45	0.06	0.12	0.24	0.37
	m=20	0.67	0.76	0.77	0.75	0.16	0.28	0.42	0.49	0.08	0.15	0.28	0.41
C=15 nPSO1	m=5	0.33	0.47	0.59	0.62	0.06	0.11	0.25	0.38	0.03	0.06	0.17	0.33
	m=10	0.58	0.69	0.73	0.72	0.12	0.21	0.36	0.46	0.06	0.11	0.24	0.38
	m=15	0.71	0.79	0.80	0.78	0.18	0.30	0.45	0.53	0.08	0.16	0.31	0.43
	m=20	0.78	0.84	0.84	0.82	0.24	0.39	0.51	0.57	0.11	0.21	0.36	0.46
C=15 nPSO2	m=5	0.29	0.43	0.50	0.55	0.06	0.11	0.23	0.35	0.03	0.05	0.15	0.29
	m=10	0.50	0.62	0.68	0.68	0.11	0.20	0.32	0.42	0.06	0.11	0.23	0.33
	m=15	0.62	0.75	0.73	0.72	0.16	0.28	0.38	0.47	0.08	0.15	0.29	0.38
	m=20	0.71	0.79	0.80	0.75	0.21	0.34	0.47	0.52	0.11	0.20	0.32	0.41
C=15 nPSO3	m=5	0.35	0.49	0.58	0.62	0.06	0.11	0.25	0.40	0.03	0.06	0.17	0.33
	m=10	0.58	0.70	0.73	0.72	0.13	0.22	0.37	0.48	0.06	0.12	0.25	0.39
	m=15	0.71	0.79	0.80	0.79	0.19	0.31	0.46	0.53	0.09	0.17	0.32	0.44
	m=20	0.78	0.84	0.84	0.82	0.25	0.39	0.52	0.58	0.12	0.22	0.37	0.47
C=20 nPSO1	m=5	0.44	0.58	0.66	0.68	0.08	0.14	0.30	0.43	0.04	0.07	0.20	0.36
	m=10	0.67	0.76	0.79	0.78	0.16	0.28	0.43	0.52	0.08	0.14	0.29	0.43
	m=15	0.78	0.84	0.85	0.83	0.24	0.39	0.51	0.58	0.12	0.21	0.36	0.48
	m=20	0.84	0.88	0.87	0.86	0.33	0.48	0.59	0.63	0.15	0.27	0.42	0.52
C=20 nPSO2	m=5	0.35	0.51	0.59	0.62	0.08	0.14	0.26	0.39	0.04	0.07	0.19	0.33
	m=10	0.60	0.73	0.74	0.71	0.14	0.25	0.38	0.46	0.08	0.14	0.26	0.39
	m=15	0.70	0.80	0.81	0.78	0.21	0.34	0.47	0.52	0.11	0.19	0.32	0.43
	m=20	0.78	0.84	0.85	0.82	0.28	0.44	0.53	0.58	0.14	0.24	0.38	0.47
C=20 nPSO3	m=5	0.44	0.60	0.66	0.69	0.09	0.16	0.31	0.45	0.05	0.08	0.21	0.37
	m=10	0.67	0.77	0.79	0.77	0.17	0.29	0.44	0.53	0.08	0.15	0.30	0.44
	m=15	0.79	0.84	0.85	0.83	0.25	0.39	0.53	0.59	0.12	0.22	0.38	0.49
	m=20	0.84	0.88	0.88	0.87	0.33	0.48	0.59	0.63	0.16	0.28	0.43	0.53

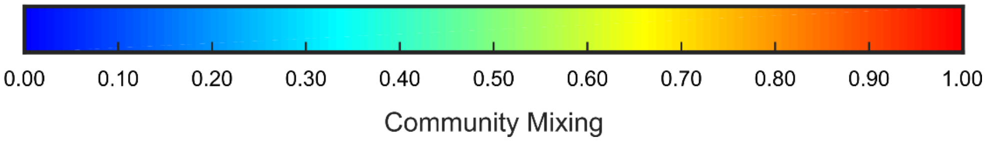
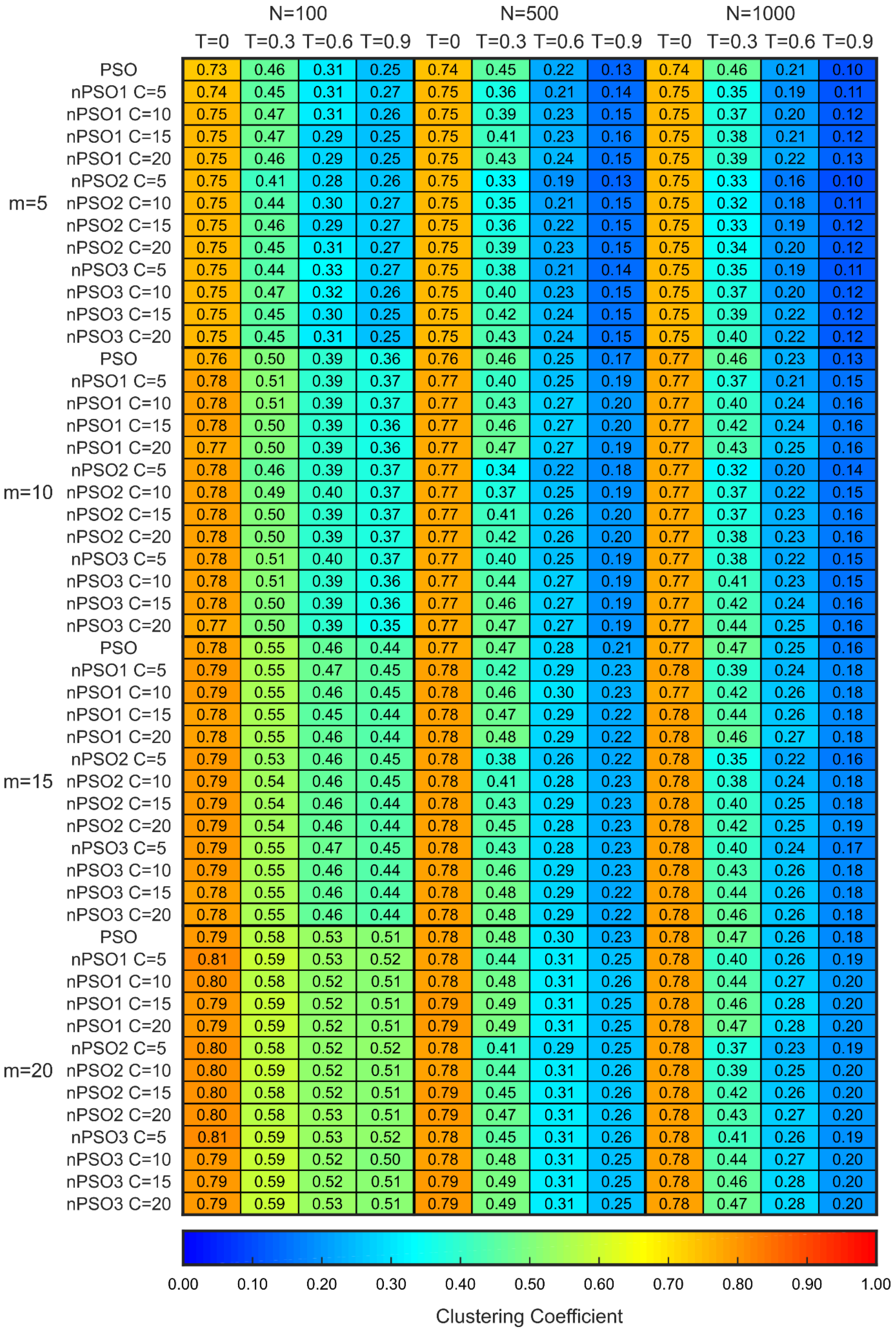


Figure 5. Community mixing on nPSO networks.

Synthetic networks have been generated using the nPSO model with parameters $N = [100, 500, 1000]$, $m = [5, 10, 15, 20]$, $T = [0, 0.3, 0.6, 0.9]$, $\gamma = 3$ and angular coordinates sampled according to mixture distributions of three different kinds with components $C = [5, 10, 15, 20]$: Gaussian mixture with equal proportions (nPSO1), Gaussian mixture with random proportions (nPSO2), Gaussian and Gamma mixture with equal proportions (nPSO3). For more details on the parameters of the mixture distributions please refer to the Methods. For each combination of parameters, 10 networks have been generated and the community mixing has been computed. The heatmap reports for each parameter combination the mean community mixing, coloured according to a blue-to-red colormap in the range [0, 1].

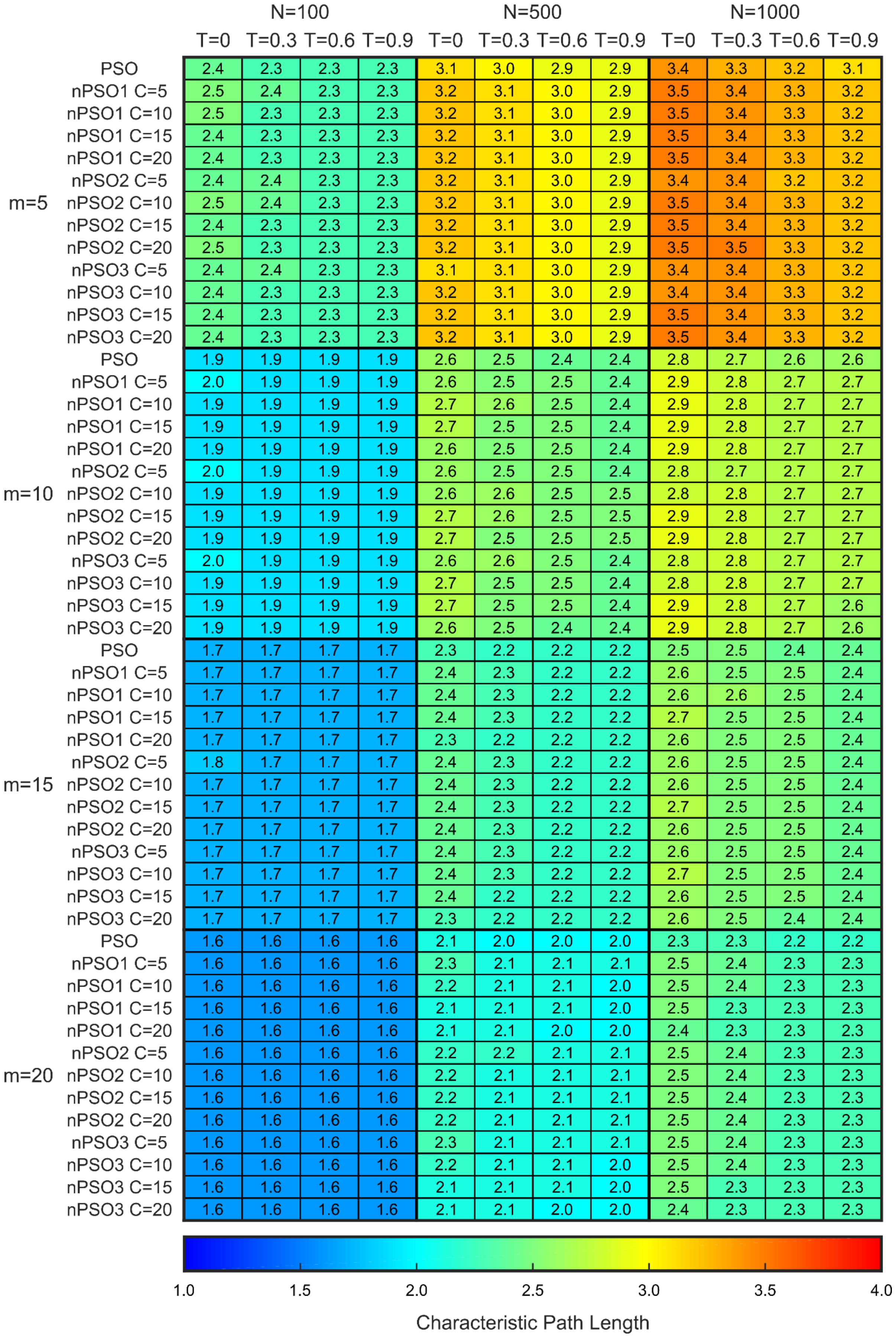


0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
Clustering Coefficient

		N=100				N=500				N=1000			
		T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9
m=5	PSO	0.73	0.46	0.31	0.25	0.74	0.45	0.22	0.13	0.74	0.46	0.21	0.10
	nPSO1 C=5	0.74	0.45	0.31	0.27	0.75	0.36	0.21	0.14	0.75	0.35	0.19	0.11
	nPSO1 C=10	0.75	0.47	0.31	0.26	0.75	0.39	0.23	0.15	0.75	0.37	0.20	0.12
	nPSO1 C=15	0.75	0.47	0.29	0.25	0.75	0.41	0.23	0.16	0.75	0.38	0.21	0.12
	nPSO1 C=20	0.75	0.46	0.29	0.25	0.75	0.43	0.24	0.15	0.75	0.39	0.22	0.13
	nPSO2 C=5	0.75	0.41	0.28	0.26	0.75	0.33	0.19	0.13	0.75	0.33	0.16	0.10
	nPSO2 C=10	0.75	0.44	0.30	0.27	0.75	0.35	0.21	0.15	0.75	0.32	0.18	0.11
	nPSO2 C=15	0.75	0.46	0.29	0.27	0.75	0.36	0.22	0.15	0.75	0.33	0.19	0.12
	nPSO2 C=20	0.75	0.45	0.31	0.27	0.75	0.39	0.23	0.15	0.75	0.34	0.20	0.12
	nPSO3 C=5	0.75	0.44	0.33	0.27	0.75	0.38	0.21	0.14	0.75	0.35	0.19	0.11
	nPSO3 C=10	0.75	0.47	0.32	0.26	0.75	0.40	0.23	0.15	0.75	0.37	0.20	0.12
	nPSO3 C=15	0.75	0.45	0.30	0.25	0.75	0.42	0.24	0.15	0.75	0.39	0.22	0.12
	nPSO3 C=20	0.75	0.45	0.31	0.25	0.75	0.43	0.24	0.15	0.75	0.40	0.22	0.12
m=10	PSO	0.76	0.50	0.39	0.36	0.76	0.46	0.25	0.17	0.77	0.46	0.23	0.13
	nPSO1 C=5	0.78	0.51	0.39	0.37	0.77	0.40	0.25	0.19	0.77	0.37	0.21	0.15
	nPSO1 C=10	0.78	0.51	0.39	0.37	0.77	0.43	0.27	0.20	0.77	0.40	0.24	0.16
	nPSO1 C=15	0.78	0.50	0.39	0.36	0.77	0.46	0.27	0.20	0.77	0.42	0.24	0.16
	nPSO1 C=20	0.77	0.50	0.39	0.36	0.77	0.47	0.27	0.19	0.77	0.43	0.25	0.16
	nPSO2 C=5	0.78	0.46	0.39	0.37	0.77	0.34	0.22	0.18	0.77	0.32	0.20	0.14
	nPSO2 C=10	0.78	0.49	0.40	0.37	0.77	0.37	0.25	0.19	0.77	0.37	0.22	0.15
	nPSO2 C=15	0.78	0.50	0.39	0.37	0.77	0.41	0.26	0.20	0.77	0.37	0.23	0.16
	nPSO2 C=20	0.78	0.50	0.39	0.37	0.77	0.42	0.26	0.20	0.77	0.38	0.23	0.16
	nPSO3 C=5	0.78	0.51	0.40	0.37	0.77	0.40	0.25	0.19	0.77	0.38	0.22	0.15
	nPSO3 C=10	0.78	0.51	0.39	0.36	0.77	0.44	0.27	0.19	0.77	0.41	0.23	0.15
	nPSO3 C=15	0.78	0.50	0.39	0.36	0.77	0.46	0.27	0.19	0.77	0.42	0.24	0.16
	nPSO3 C=20	0.77	0.50	0.39	0.35	0.77	0.47	0.27	0.19	0.77	0.44	0.25	0.16
m=15	PSO	0.78	0.55	0.46	0.44	0.77	0.47	0.28	0.21	0.77	0.47	0.25	0.16
	nPSO1 C=5	0.79	0.55	0.47	0.45	0.78	0.42	0.29	0.23	0.78	0.39	0.24	0.18
	nPSO1 C=10	0.79	0.55	0.46	0.45	0.78	0.46	0.30	0.23	0.77	0.42	0.26	0.18
	nPSO1 C=15	0.78	0.55	0.45	0.44	0.78	0.47	0.29	0.22	0.78	0.44	0.26	0.18
	nPSO1 C=20	0.78	0.55	0.46	0.44	0.78	0.48	0.29	0.22	0.78	0.46	0.27	0.18
	nPSO2 C=5	0.79	0.53	0.46	0.45	0.78	0.38	0.26	0.22	0.78	0.35	0.22	0.16
	nPSO2 C=10	0.79	0.54	0.46	0.45	0.78	0.41	0.28	0.23	0.78	0.38	0.24	0.18
	nPSO2 C=15	0.79	0.54	0.46	0.44	0.78	0.43	0.29	0.23	0.78	0.40	0.25	0.18
	nPSO2 C=20	0.79	0.54	0.46	0.44	0.78	0.45	0.28	0.23	0.78	0.42	0.25	0.19
	nPSO3 C=5	0.79	0.55	0.47	0.45	0.78	0.43	0.28	0.23	0.78	0.40	0.24	0.17
	nPSO3 C=10	0.79	0.55	0.46	0.44	0.78	0.46	0.29	0.23	0.78	0.43	0.26	0.18
	nPSO3 C=15	0.78	0.55	0.46	0.44	0.78	0.48	0.29	0.22	0.78	0.44	0.26	0.18
	nPSO3 C=20	0.78	0.55	0.46	0.44	0.78	0.48	0.29	0.22	0.78	0.46	0.26	0.18
m=20	PSO	0.79	0.58	0.53	0.51	0.78	0.48	0.30	0.23	0.78	0.47	0.26	0.18
	nPSO1 C=5	0.81	0.59	0.53	0.52	0.78	0.44	0.31	0.25	0.78	0.40	0.26	0.19
	nPSO1 C=10	0.80	0.58	0.52	0.51	0.78	0.48	0.31	0.26	0.78	0.44	0.27	0.20
	nPSO1 C=15	0.79	0.59	0.52	0.51	0.79	0.49	0.31	0.25	0.78	0.46	0.28	0.20
	nPSO1 C=20	0.79	0.59	0.52	0.51	0.79	0.49	0.31	0.25	0.78	0.47	0.28	0.20
	nPSO2 C=5	0.80	0.58	0.52	0.52	0.78	0.41	0.29	0.25	0.78	0.37	0.23	0.19
	nPSO2 C=10	0.80	0.59	0.52	0.51	0.78	0.44	0.31	0.26	0.78	0.39	0.25	0.20
	nPSO2 C=15	0.80	0.58	0.52	0.51	0.79	0.45	0.31	0.26	0.78	0.42	0.26	0.20
	nPSO2 C=20	0.80	0.58	0.53	0.51	0.79	0.47	0.31	0.26	0.78	0.43	0.27	0.20
	nPSO3 C=5	0.81	0.59	0.53	0.52	0.78	0.45	0.31	0.26	0.78	0.41	0.26	0.19
	nPSO3 C=10	0.79	0.59	0.52	0.50	0.78	0.48	0.31	0.25	0.78	0.44	0.27	0.20
	nPSO3 C=15	0.79	0.59	0.52	0.51	0.79	0.49	0.31	0.25	0.78	0.46	0.28	0.20
	nPSO3 C=20	0.79	0.59	0.53	0.51	0.79	0.49	0.31	0.25	0.78	0.47	0.28	0.20

Figure 6. Clustering coefficient of the PSO and nPSO networks.

Synthetic networks have been generated using the nPSO model with parameters N = [100, 500, 1000], m = [5, 10, 15, 20], T = [0, 0.3, 0.6, 0.9], γ = 3 and angular coordinates sampled according to mixture distributions of three different kinds with components C = [5, 10, 15, 20]: Gaussian mixture with equal proportions (nPSO1), Gaussian mixture with random proportions (nPSO2), Gaussian and Gamma mixture with equal proportions (nPSO3). For more details on the parameters of the mixture distributions please refer to the Methods. Furthermore, synthetic networks have been generated also using the PSO model with the same parameters N, m, T and γ. For each combination of parameters, 10 networks have been generated and the clustering coefficient has been computed. The heatmap reports for each parameter combination the mean clustering coefficient, coloured according to a blue-to-red colormap in the range [0, 1].



		N=100				N=500				N=1000			
		T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9
m=5	PSO	2.4	2.3	2.3	2.3	3.1	3.0	2.9	2.9	3.4	3.3	3.2	3.1
	nPSO1 C=5	2.5	2.4	2.3	2.3	3.2	3.1	3.0	2.9	3.5	3.4	3.3	3.2
	nPSO1 C=10	2.5	2.3	2.3	2.3	3.2	3.1	3.0	2.9	3.5	3.4	3.3	3.2
	nPSO1 C=15	2.4	2.3	2.3	2.3	3.2	3.1	3.0	2.9	3.5	3.4	3.3	3.2
	nPSO1 C=20	2.4	2.3	2.3	2.3	3.2	3.1	3.0	2.9	3.5	3.4	3.3	3.2
	nPSO2 C=5	2.4	2.4	2.3	2.3	3.2	3.1	3.0	2.9	3.4	3.4	3.2	3.2
	nPSO2 C=10	2.5	2.4	2.3	2.3	3.2	3.1	3.0	2.9	3.5	3.4	3.3	3.2
	nPSO2 C=15	2.4	2.3	2.3	2.3	3.2	3.1	3.0	2.9	3.5	3.4	3.3	3.2
	nPSO2 C=20	2.5	2.3	2.3	2.3	3.2	3.1	3.0	2.9	3.5	3.5	3.3	3.2
	nPSO3 C=5	2.4	2.4	2.3	2.3	3.1	3.1	3.0	2.9	3.4	3.4	3.3	3.2
	nPSO3 C=10	2.4	2.3	2.3	2.3	3.2	3.1	3.0	2.9	3.4	3.4	3.3	3.2
	nPSO3 C=15	2.4	2.3	2.3	2.3	3.2	3.1	3.0	2.9	3.5	3.4	3.3	3.2
	nPSO3 C=20	2.4	2.3	2.3	2.3	3.2	3.1	3.0	2.9	3.5	3.4	3.3	3.2
m=10	PSO	1.9	1.9	1.9	1.9	2.6	2.5	2.4	2.4	2.8	2.7	2.6	2.6
	nPSO1 C=5	2.0	1.9	1.9	1.9	2.6	2.5	2.5	2.4	2.9	2.8	2.7	2.7
	nPSO1 C=10	1.9	1.9	1.9	1.9	2.7	2.6	2.5	2.4	2.9	2.8	2.7	2.7
	nPSO1 C=15	1.9	1.9	1.9	1.9	2.7	2.5	2.5	2.4	2.9	2.8	2.7	2.7
	nPSO1 C=20	1.9	1.9	1.9	1.9	2.6	2.5	2.5	2.4	2.9	2.8	2.7	2.7
	nPSO2 C=5	2.0	1.9	1.9	1.9	2.6	2.5	2.5	2.4	2.8	2.7	2.7	2.7
	nPSO2 C=10	1.9	1.9	1.9	1.9	2.6	2.6	2.5	2.5	2.8	2.8	2.7	2.7
	nPSO2 C=15	1.9	1.9	1.9	1.9	2.7	2.6	2.5	2.5	2.9	2.8	2.7	2.7
	nPSO2 C=20	1.9	1.9	1.9	1.9	2.7	2.5	2.5	2.5	2.9	2.8	2.7	2.7
	nPSO3 C=5	2.0	1.9	1.9	1.9	2.6	2.6	2.5	2.4	2.8	2.8	2.7	2.7
	nPSO3 C=10	1.9	1.9	1.9	1.9	2.7	2.5	2.5	2.4	2.8	2.8	2.7	2.7
	nPSO3 C=15	1.9	1.9	1.9	1.9	2.7	2.5	2.5	2.4	2.9	2.8	2.7	2.6
	nPSO3 C=20	1.9	1.9	1.9	1.9	2.6	2.5	2.4	2.4	2.9	2.8	2.7	2.6
m=15	PSO	1.7	1.7	1.7	1.7	2.3	2.2	2.2	2.2	2.5	2.5	2.4	2.4
	nPSO1 C=5	1.7	1.7	1.7	1.7	2.4	2.3	2.2	2.2	2.6	2.5	2.5	2.4
	nPSO1 C=10	1.7	1.7	1.7	1.7	2.4	2.3	2.2	2.2	2.6	2.6	2.5	2.4
	nPSO1 C=15	1.7	1.7	1.7	1.7	2.4	2.3	2.2	2.2	2.7	2.5	2.5	2.4
	nPSO1 C=20	1.7	1.7	1.7	1.7	2.3	2.2	2.2	2.2	2.6	2.5	2.5	2.4
	nPSO2 C=5	1.8	1.7	1.7	1.7	2.4	2.3	2.2	2.2	2.6	2.5	2.5	2.4
	nPSO2 C=10	1.7	1.7	1.7	1.7	2.4	2.3	2.2	2.2	2.6	2.5	2.5	2.4
	nPSO2 C=15	1.7	1.7	1.7	1.7	2.4	2.3	2.2	2.2	2.7	2.5	2.5	2.4
	nPSO2 C=20	1.7	1.7	1.7	1.7	2.4	2.3	2.2	2.2	2.6	2.5	2.5	2.4
	nPSO3 C=5	1.7	1.7	1.7	1.7	2.4	2.3	2.2	2.2	2.6	2.5	2.5	2.4
	nPSO3 C=10	1.7	1.7	1.7	1.7	2.4	2.3	2.2	2.2	2.7	2.5	2.5	2.4
	nPSO3 C=15	1.7	1.7	1.7	1.7	2.4	2.2	2.2	2.2	2.6	2.5	2.5	2.4
	nPSO3 C=20	1.7	1.7	1.7	1.7	2.3	2.2	2.2	2.2	2.6	2.5	2.4	2.4
m=20	PSO	1.6	1.6	1.6	1.6	2.1	2.0	2.0	2.0	2.3	2.3	2.2	2.2
	nPSO1 C=5	1.6	1.6	1.6	1.6	2.3	2.1	2.1	2.1	2.5	2.4	2.3	2.3
	nPSO1 C=10	1.6	1.6	1.6	1.6	2.2	2.1	2.1	2.0	2.5	2.4	2.3	2.3
	nPSO1 C=15	1.6	1.6	1.6	1.6	2.1	2.1	2.1	2.0	2.5	2.3	2.3	2.3
	nPSO1 C=20	1.6	1.6	1.6	1.6	2.1	2.1	2.0	2.0	2.4	2.3	2.3	2.3
	nPSO2 C=5	1.6	1.6	1.6	1.6	2.2	2.2	2.1	2.1	2.5	2.4	2.3	2.3
	nPSO2 C=10	1.6	1.6	1.6	1.6	2.2	2.1	2.1	2.1	2.5	2.4	2.3	2.3
	nPSO2 C=15	1.6	1.6	1.6	1.6	2.2	2.1	2.1	2.1	2.5	2.4	2.3	2.3
	nPSO2 C=20	1.6	1.6	1.6	1.6	2.2	2.1	2.1	2.1	2.5	2.4	2.3	2.3
	nPSO3 C=5	1.6	1.6	1.6	1.6	2.3	2.1	2.1	2.1	2.5	2.4	2.3	2.3
	nPSO3 C=10	1.6	1.6	1.6	1.6	2.2	2.1	2.1	2.0	2.5	2.4	2.3	2.3
	nPSO3 C=15	1.6	1.6	1.6	1.6	2.1	2.1	2.1	2.0	2.5	2.3	2.3	2.3
	nPSO3 C=20	1.6	1.6	1.6	1.6	2.1	2.1	2.0	2.0	2.4	2.3	2.3	2.3

Figure 7. Characteristic path length of the PSO and nPSO networks.

Synthetic networks have been generated using the nPSO model with parameters N = [100, 500, 1000], m = [5, 10, 15, 20], T = [0, 0.3, 0.6, 0.9], γ = 3 and angular coordinates sampled according to mixture distributions of three different kinds with components C = [5, 10, 15, 20]: Gaussian mixture with equal proportions (nPSO1), Gaussian mixture with random proportions (nPSO2), Gaussian and Gamma mixture with equal proportions (nPSO3). For more details on the parameters of the mixture distributions please refer to the Methods. Furthermore, synthetic networks have been generated also using the PSO model with the same parameters N, m, T and γ. For each combination of parameters, 10 networks have been generated and the characteristic path length has been computed. The heatmap reports for each parameter combination the mean characteristic path length, coloured according to a blue-to-red colormap in the range [1, 4].

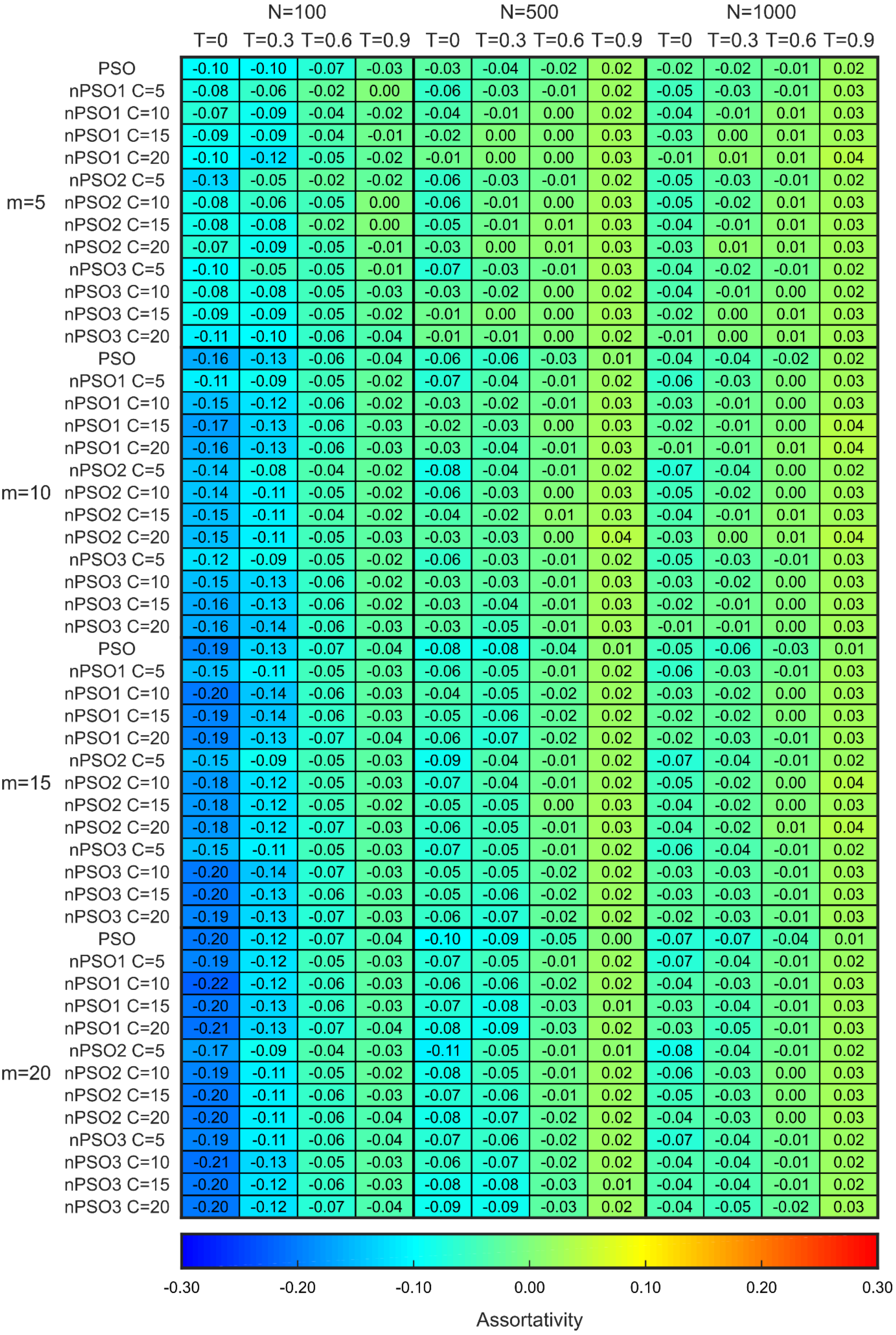
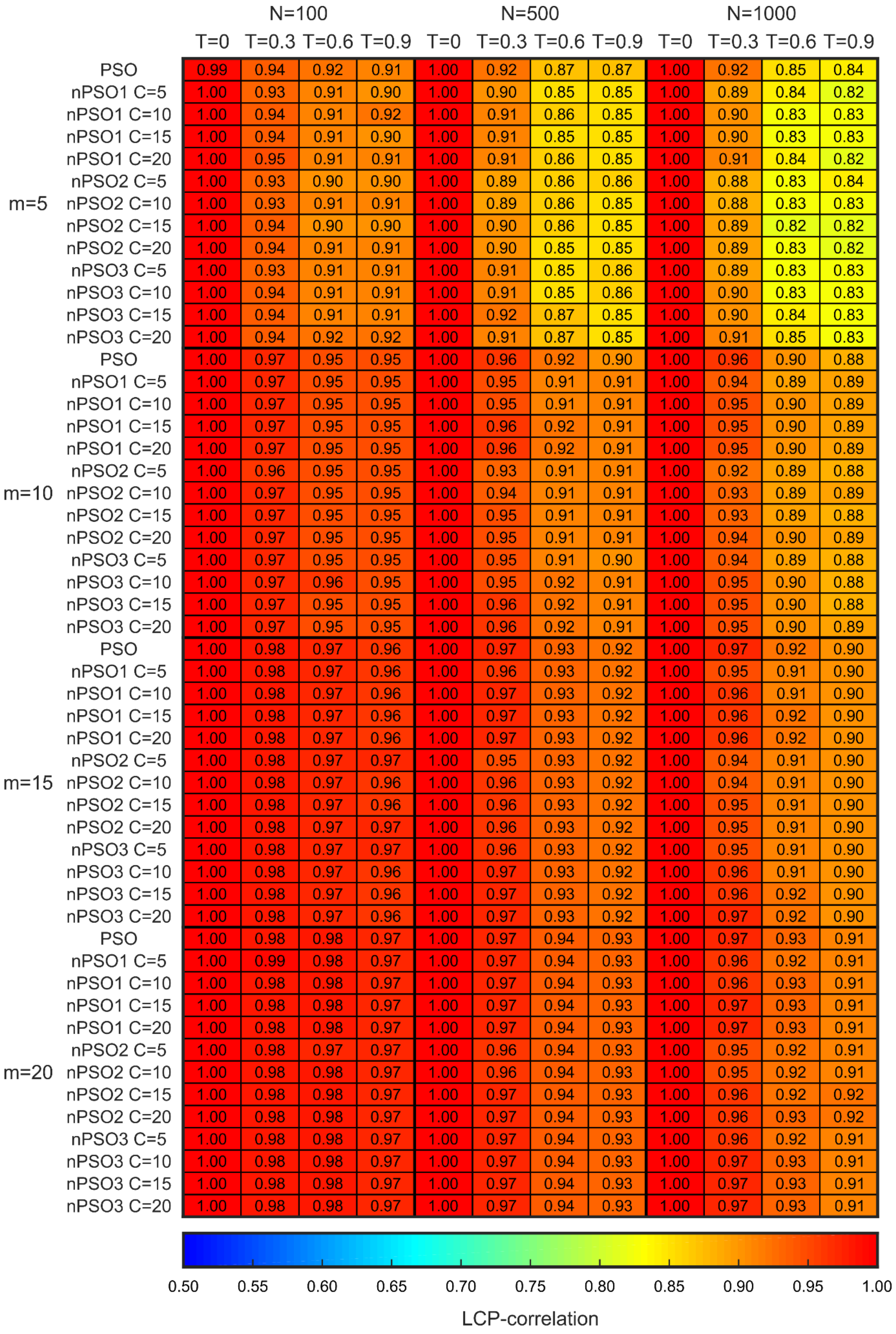


Figure 8. Assortativity of the PSO and nPSO networks.

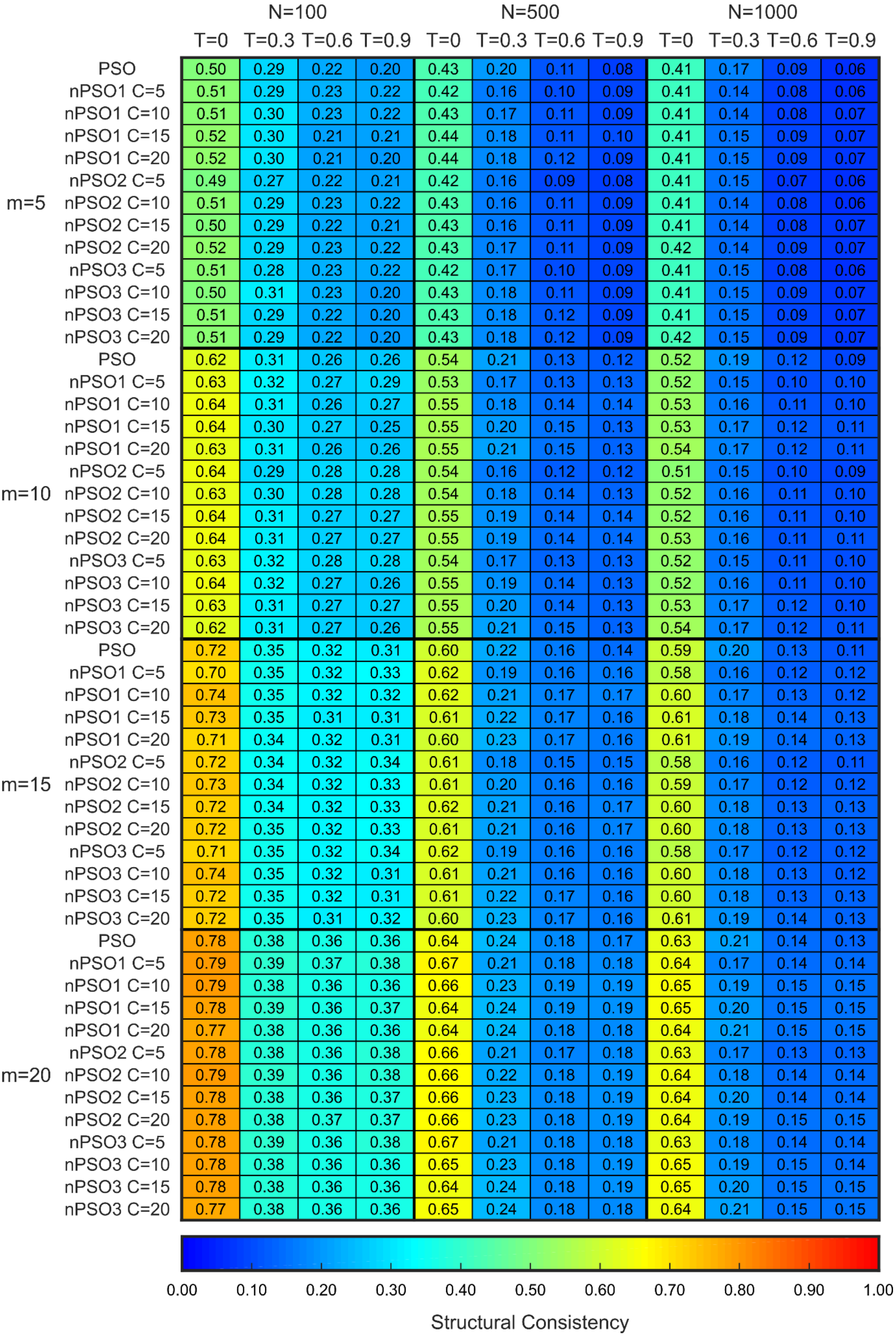
Synthetic networks have been generated using the nPSO model with parameters *N* = [100, 500, 1000], *m* = [5, 10, 15, 20], *T* = [0, 0.3, 0.6, 0.9], *γ* = 3 and angular coordinates sampled according to mixture distributions of three different kinds with components *C* = [5, 10, 15, 20]: Gaussian mixture with equal proportions (nPSO1), Gaussian mixture with random proportions (nPSO2), Gaussian and Gamma mixture with equal proportions (nPSO3). For more details on the parameters of the mixture distributions please refer to the Methods. Furthermore, synthetic networks have been generated also using the PSO model with the same parameters *N*, *m*, *T* and *γ*. For each combination of parameters, 10 networks have been generated and the assortativity has been computed. The heatmap reports for each parameter combination the mean assortativity, coloured according to a blue-to-red colormap in the range [-0.3, 0.3].



		N=100				N=500				N=1000			
		T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9
m=5	PSO	0.99	0.94	0.92	0.91	1.00	0.92	0.87	0.87	1.00	0.92	0.85	0.84
	nPSO1 C=5	1.00	0.93	0.91	0.90	1.00	0.90	0.85	0.85	1.00	0.89	0.84	0.82
	nPSO1 C=10	1.00	0.94	0.91	0.92	1.00	0.91	0.86	0.85	1.00	0.90	0.83	0.83
	nPSO1 C=15	1.00	0.94	0.91	0.90	1.00	0.91	0.85	0.85	1.00	0.90	0.83	0.83
	nPSO1 C=20	1.00	0.95	0.91	0.91	1.00	0.91	0.86	0.85	1.00	0.91	0.84	0.82
	nPSO2 C=5	1.00	0.93	0.90	0.90	1.00	0.89	0.86	0.86	1.00	0.88	0.83	0.84
	nPSO2 C=10	1.00	0.93	0.91	0.91	1.00	0.89	0.86	0.85	1.00	0.88	0.83	0.83
	nPSO2 C=15	1.00	0.94	0.90	0.90	1.00	0.90	0.86	0.85	1.00	0.89	0.82	0.82
	nPSO2 C=20	1.00	0.94	0.91	0.91	1.00	0.90	0.85	0.85	1.00	0.89	0.83	0.82
	nPSO3 C=5	1.00	0.93	0.91	0.91	1.00	0.91	0.85	0.86	1.00	0.89	0.83	0.83
	nPSO3 C=10	1.00	0.94	0.91	0.91	1.00	0.91	0.85	0.86	1.00	0.90	0.83	0.83
	nPSO3 C=15	1.00	0.94	0.91	0.91	1.00	0.92	0.87	0.85	1.00	0.90	0.84	0.83
	nPSO3 C=20	1.00	0.94	0.92	0.92	1.00	0.91	0.87	0.85	1.00	0.91	0.85	0.83
m=10	PSO	1.00	0.97	0.95	0.95	1.00	0.96	0.92	0.90	1.00	0.96	0.90	0.88
	nPSO1 C=5	1.00	0.97	0.95	0.95	1.00	0.95	0.91	0.91	1.00	0.94	0.89	0.89
	nPSO1 C=10	1.00	0.97	0.95	0.95	1.00	0.95	0.91	0.91	1.00	0.95	0.90	0.89
	nPSO1 C=15	1.00	0.97	0.95	0.95	1.00	0.96	0.92	0.91	1.00	0.95	0.90	0.89
	nPSO1 C=20	1.00	0.97	0.95	0.95	1.00	0.96	0.92	0.91	1.00	0.95	0.90	0.89
	nPSO2 C=5	1.00	0.96	0.95	0.95	1.00	0.93	0.91	0.91	1.00	0.92	0.89	0.88
	nPSO2 C=10	1.00	0.97	0.95	0.95	1.00	0.94	0.91	0.91	1.00	0.93	0.89	0.89
	nPSO2 C=15	1.00	0.97	0.95	0.95	1.00	0.95	0.91	0.91	1.00	0.93	0.89	0.88
	nPSO2 C=20	1.00	0.97	0.95	0.95	1.00	0.95	0.91	0.91	1.00	0.94	0.90	0.89
	nPSO3 C=5	1.00	0.97	0.95	0.95	1.00	0.95	0.91	0.90	1.00	0.94	0.89	0.88
	nPSO3 C=10	1.00	0.97	0.96	0.95	1.00	0.95	0.92	0.91	1.00	0.95	0.90	0.88
	nPSO3 C=15	1.00	0.97	0.95	0.95	1.00	0.96	0.92	0.91	1.00	0.95	0.90	0.88
	nPSO3 C=20	1.00	0.97	0.95	0.95	1.00	0.96	0.92	0.91	1.00	0.95	0.90	0.89
m=15	PSO	1.00	0.98	0.97	0.96	1.00	0.97	0.93	0.92	1.00	0.97	0.92	0.90
	nPSO1 C=5	1.00	0.98	0.97	0.96	1.00	0.96	0.93	0.92	1.00	0.95	0.91	0.90
	nPSO1 C=10	1.00	0.98	0.97	0.96	1.00	0.97	0.93	0.92	1.00	0.96	0.91	0.90
	nPSO1 C=15	1.00	0.98	0.97	0.96	1.00	0.97	0.93	0.92	1.00	0.96	0.92	0.90
	nPSO1 C=20	1.00	0.98	0.97	0.96	1.00	0.97	0.93	0.92	1.00	0.96	0.92	0.90
	nPSO2 C=5	1.00	0.98	0.97	0.97	1.00	0.95	0.93	0.92	1.00	0.94	0.91	0.90
	nPSO2 C=10	1.00	0.98	0.97	0.96	1.00	0.96	0.93	0.92	1.00	0.94	0.91	0.90
	nPSO2 C=15	1.00	0.98	0.97	0.96	1.00	0.96	0.93	0.92	1.00	0.95	0.91	0.90
	nPSO2 C=20	1.00	0.98	0.97	0.97	1.00	0.96	0.93	0.92	1.00	0.95	0.91	0.90
	nPSO3 C=5	1.00	0.98	0.97	0.97	1.00	0.96	0.93	0.92	1.00	0.95	0.91	0.90
	nPSO3 C=10	1.00	0.98	0.97	0.96	1.00	0.97	0.93	0.92	1.00	0.96	0.91	0.90
	nPSO3 C=15	1.00	0.98	0.97	0.96	1.00	0.97	0.93	0.92	1.00	0.96	0.92	0.90
	nPSO3 C=20	1.00	0.98	0.97	0.96	1.00	0.97	0.93	0.92	1.00	0.97	0.92	0.90
m=20	PSO	1.00	0.98	0.98	0.97	1.00	0.97	0.94	0.93	1.00	0.97	0.93	0.91
	nPSO1 C=5	1.00	0.99	0.98	0.97	1.00	0.97	0.94	0.93	1.00	0.96	0.92	0.91
	nPSO1 C=10	1.00	0.98	0.98	0.97	1.00	0.97	0.94	0.93	1.00	0.96	0.93	0.91
	nPSO1 C=15	1.00	0.98	0.98	0.97	1.00	0.97	0.94	0.93	1.00	0.97	0.93	0.91
	nPSO1 C=20	1.00	0.98	0.98	0.97	1.00	0.97	0.94	0.93	1.00	0.97	0.93	0.91
	nPSO2 C=5	1.00	0.98	0.97	0.97	1.00	0.96	0.94	0.93	1.00	0.95	0.92	0.91
	nPSO2 C=10	1.00	0.98	0.98	0.97	1.00	0.96	0.94	0.93	1.00	0.95	0.92	0.91
	nPSO2 C=15	1.00	0.98	0.98	0.97	1.00	0.97	0.94	0.93	1.00	0.96	0.92	0.92
	nPSO2 C=20	1.00	0.98	0.98	0.97	1.00	0.97	0.94	0.93	1.00	0.96	0.93	0.92
	nPSO3 C=5	1.00	0.98	0.98	0.97	1.00	0.97	0.94	0.93	1.00	0.96	0.92	0.91
	nPSO3 C=10	1.00	0.98	0.98	0.97	1.00	0.97	0.94	0.93	1.00	0.97	0.93	0.91
	nPSO3 C=15	1.00	0.98	0.98	0.97	1.00	0.97	0.94	0.93	1.00	0.97	0.93	0.91
	nPSO3 C=20	1.00	0.98	0.98	0.97	1.00	0.97	0.94	0.93	1.00	0.97	0.93	0.91

Figure 9. LCP-correlation of the PSO and nPSO networks.

Synthetic networks have been generated using the nPSO model with parameters $N = [100, 500, 1000]$, $m = [5, 10, 15, 20]$, $T = [0, 0.3, 0.6, 0.9]$, $\gamma = 3$ and angular coordinates sampled according to mixture distributions of three different kinds with components $C = [5, 10, 15, 20]$: Gaussian mixture with equal proportions (nPSO1), Gaussian mixture with random proportions (nPSO2), Gaussian and Gamma mixture with equal proportions (nPSO3). For more details on the parameters of the mixture distributions please refer to the Methods. Furthermore, synthetic networks have been generated also using the PSO model with the same parameters N, m, T and γ. For each combination of parameters, 10 networks have been generated and the LCP-correlation has been computed. The heatmap reports for each parameter combination the mean LCP-correlation, coloured according to a blue-to-red colormap in the range [0.5, 1].



		N=100				N=500				N=1000			
		T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9
m=5	PSO	0.50	0.29	0.22	0.20	0.43	0.20	0.11	0.08	0.41	0.17	0.09	0.06
	nPSO1 C=5	0.51	0.29	0.23	0.22	0.42	0.16	0.10	0.09	0.41	0.14	0.08	0.06
	nPSO1 C=10	0.51	0.30	0.23	0.22	0.43	0.17	0.11	0.09	0.41	0.14	0.08	0.07
	nPSO1 C=15	0.52	0.30	0.21	0.21	0.44	0.18	0.11	0.10	0.41	0.15	0.09	0.07
	nPSO1 C=20	0.52	0.30	0.21	0.20	0.44	0.18	0.12	0.09	0.41	0.15	0.09	0.07
	nPSO2 C=5	0.49	0.27	0.22	0.21	0.42	0.16	0.09	0.08	0.41	0.15	0.07	0.06
	nPSO2 C=10	0.51	0.29	0.23	0.22	0.43	0.16	0.11	0.09	0.41	0.14	0.08	0.06
	nPSO2 C=15	0.50	0.29	0.22	0.21	0.43	0.16	0.11	0.09	0.41	0.14	0.08	0.07
	nPSO2 C=20	0.52	0.29	0.23	0.22	0.43	0.17	0.11	0.09	0.42	0.14	0.09	0.07
	nPSO3 C=5	0.51	0.28	0.23	0.22	0.42	0.17	0.10	0.09	0.41	0.15	0.08	0.06
	nPSO3 C=10	0.50	0.31	0.23	0.20	0.43	0.18	0.11	0.09	0.41	0.15	0.09	0.07
	nPSO3 C=15	0.51	0.29	0.22	0.20	0.43	0.18	0.12	0.09	0.41	0.15	0.09	0.07
	nPSO3 C=20	0.51	0.29	0.22	0.20	0.43	0.18	0.12	0.09	0.42	0.15	0.09	0.07
m=10	PSO	0.62	0.31	0.26	0.26	0.54	0.21	0.13	0.12	0.52	0.19	0.12	0.09
	nPSO1 C=5	0.63	0.32	0.27	0.29	0.53	0.17	0.13	0.13	0.52	0.15	0.10	0.10
	nPSO1 C=10	0.64	0.31	0.26	0.27	0.55	0.18	0.14	0.14	0.53	0.16	0.11	0.10
	nPSO1 C=15	0.64	0.30	0.27	0.25	0.55	0.20	0.15	0.13	0.53	0.17	0.12	0.11
	nPSO1 C=20	0.63	0.31	0.26	0.26	0.55	0.21	0.15	0.13	0.54	0.17	0.12	0.11
	nPSO2 C=5	0.64	0.29	0.28	0.28	0.54	0.16	0.12	0.12	0.51	0.15	0.10	0.09
	nPSO2 C=10	0.63	0.30	0.28	0.28	0.54	0.18	0.14	0.13	0.52	0.16	0.11	0.10
	nPSO2 C=15	0.64	0.31	0.27	0.27	0.55	0.19	0.14	0.14	0.52	0.16	0.11	0.10
	nPSO2 C=20	0.64	0.31	0.27	0.27	0.55	0.19	0.14	0.14	0.53	0.16	0.11	0.11
	nPSO3 C=5	0.63	0.32	0.28	0.28	0.54	0.17	0.13	0.13	0.52	0.15	0.11	0.10
	nPSO3 C=10	0.64	0.32	0.27	0.26	0.55	0.19	0.14	0.13	0.52	0.16	0.11	0.10
	nPSO3 C=15	0.63	0.31	0.27	0.27	0.55	0.20	0.14	0.13	0.53	0.17	0.12	0.10
	nPSO3 C=20	0.62	0.31	0.27	0.26	0.55	0.21	0.15	0.13	0.54	0.17	0.12	0.11
m=15	PSO	0.72	0.35	0.32	0.31	0.60	0.22	0.16	0.14	0.59	0.20	0.13	0.11
	nPSO1 C=5	0.70	0.35	0.32	0.33	0.62	0.19	0.16	0.16	0.58	0.16	0.12	0.12
	nPSO1 C=10	0.74	0.35	0.32	0.32	0.62	0.21	0.17	0.17	0.60	0.17	0.13	0.12
	nPSO1 C=15	0.73	0.35	0.31	0.31	0.61	0.22	0.17	0.16	0.61	0.18	0.14	0.13
	nPSO1 C=20	0.71	0.34	0.32	0.31	0.60	0.23	0.17	0.16	0.61	0.19	0.14	0.13
	nPSO2 C=5	0.72	0.34	0.32	0.34	0.61	0.18	0.15	0.15	0.58	0.16	0.12	0.11
	nPSO2 C=10	0.73	0.34	0.32	0.33	0.61	0.20	0.16	0.16	0.59	0.17	0.12	0.12
	nPSO2 C=15	0.72	0.34	0.32	0.33	0.62	0.21	0.16	0.17	0.60	0.18	0.13	0.13
	nPSO2 C=20	0.72	0.35	0.32	0.33	0.61	0.21	0.16	0.17	0.60	0.18	0.13	0.13
	nPSO3 C=5	0.71	0.35	0.32	0.34	0.62	0.19	0.16	0.16	0.58	0.17	0.12	0.12
	nPSO3 C=10	0.74	0.35	0.32	0.31	0.61	0.21	0.16	0.16	0.60	0.18	0.13	0.12
	nPSO3 C=15	0.72	0.35	0.32	0.31	0.61	0.22	0.17	0.16	0.60	0.18	0.13	0.13
	nPSO3 C=20	0.72	0.35	0.31	0.32	0.60	0.23	0.17	0.16	0.61	0.19	0.14	0.13
m=20	PSO	0.78	0.38	0.36	0.36	0.64	0.24	0.18	0.17	0.63	0.21	0.14	0.13
	nPSO1 C=5	0.79	0.39	0.37	0.38	0.67	0.21	0.18	0.18	0.64	0.17	0.14	0.14
	nPSO1 C=10	0.79	0.38	0.36	0.36	0.66	0.23	0.19	0.19	0.65	0.19	0.15	0.15
	nPSO1 C=15	0.78	0.39	0.36	0.37	0.64	0.24	0.19	0.19	0.65	0.20	0.15	0.15
	nPSO1 C=20	0.77	0.38	0.36	0.36	0.64	0.24	0.18	0.18	0.64	0.21	0.15	0.15
	nPSO2 C=5	0.78	0.38	0.36	0.38	0.66	0.21	0.17	0.18	0.63	0.17	0.13	0.13
	nPSO2 C=10	0.79	0.39	0.36	0.38	0.66	0.22	0.18	0.19	0.64	0.18	0.14	0.14
	nPSO2 C=15	0.78	0.38	0.36	0.37	0.66	0.23	0.18	0.19	0.64	0.20	0.14	0.14
	nPSO2 C=20	0.78	0.38	0.37	0.37	0.66	0.23	0.18	0.19	0.64	0.19	0.15	0.15
	nPSO3 C=5	0.78	0.39	0.36	0.38	0.67	0.21	0.18	0.18	0.63	0.18	0.14	0.14
	nPSO3 C=10	0.78	0.38	0.36	0.36	0.65	0.23	0.18	0.19	0.65	0.19	0.15	0.14
	nPSO3 C=15	0.78	0.38	0.36	0.36	0.64	0.24	0.18	0.19	0.65	0.20	0.15	0.15
	nPSO3 C=20	0.77	0.38	0.36	0.36	0.65	0.24	0.18	0.18	0.64	0.21	0.15	0.15

Figure 10. Structural consistency of the PSO and nPSO networks.

Synthetic networks have been generated using the nPSO model with parameters N = [100, 500, 1000], m = [5, 10, 15, 20], T = [0, 0.3, 0.6, 0.9], γ = 3 and angular coordinates sampled according to mixture distributions of three different kinds with components C = [5, 10, 15, 20]: Gaussian mixture with equal proportions (nPSO1), Gaussian mixture with random proportions (nPSO2), Gaussian and Gamma mixture with equal proportions (nPSO3). For more details on the parameters of the mixture distributions please refer to the Methods. Furthermore, synthetic networks have been generated also using the PSO model with the same parameters N, m, T and γ. For each combination of parameters, 10 networks have been generated and the structural consistency has been computed. The heatmap reports for each parameter combination the mean structural consistency, coloured according to a blue-to-red colormap in the range [0, 1].

B.6 Supplementary Information

Methods

Generation of Gaussian and Gamma mixture distribution

The procedure here described aims at generating a Gaussian and Gamma mixture distribution and consists in separately building a certain number *C* of components (Gaussian or Gamma) in the range [0, 2π[and then combining them. Note that the same procedure is valid also to combine only Gaussian components and can be applied to generate the Gaussian mixture distribution described in the main article, in case your programming tools do not provide already the possibility for a random sampling from a Gaussian mixture distribution.

Given a number *C* of components, we generated half of them as Gaussian and half as Gamma, with an additional one among the two at random if *C* is an odd number. The means of the components are equidistantly arranged over the angular space:

$$\mu_i = \frac{2\pi}{C} * (i - 1) \quad i = 1 \dots C$$

The ordering of the components is chosen at random.

For each Gaussian component, the parameters to set are the mean, fixed to $\mu = \mu_i$, and the standard deviation, fixed to 1/6 of the distance between two adjacent means $\left(\sigma = \frac{1}{6} * \frac{2\pi}{C}\right)$, which allowed for a reasonable isolation of the communities independently from their number.

For each Gamma component, the parameters to set are the shape parameter, fixed to $a = 2$, and the scale parameter, fixed to $b = 1/C$, which allowed for a reasonable isolation of the communities independently from their number. Note that the mean of the distribution is given by the product of the two parameters $a * b$.

For each component i, given the type of distribution and the parameters, we generate a probability density function Y_i evaluated at a sufficiently high number (i.e. 10^4) of evenly spaced points *X* in [0, 2π[. Since a relevant part of the probability density function might lie out of this range, we iteratively repeat the evaluation also for points X^* greater than 2π ([2π, 4π[, [4π, 6π[, …) and lower than 0 ([-2π, 0[, [-4π, -2π[, …), and in both the cases we stop when the values of the probability density function are lower than a tolerance threshold (i.e. 10^{-4}). The related values Y_i^* are added up to the initial Y_i.

If the component is a Gamma distribution, further operations has to be performed. While for the Gaussian distribution changing the mean parameter allows to shift the curve toward the desired mean μ_i preserving the shape, for the Gamma distribution this is not valid, since fixing the mean by changing the parameters a and b also results in a change of shape. Therefore, we keep fixed the parameters a and b and we perform a translation of Y_i such that the expected mean $a * b$ is shifted toward the desired mean μ_i. Note that while performing the translation the range [0, 2π[has to be considered in a circular way, therefore the translated Y_i is still defined for the same points X as before.

Furthermore, the Gamma distribution is not symmetric with respect to its mean, in fact the mode is located to the left of the mean and in correspondence to $(a - 1) * b$ for $a \geq 1$. In order to increase the possible asymmetries in the final mixture distribution, as a random alternative we perform a reflection of Y_i with respect to its mean, so that the mode would be located to the right of the mean. Note that while performing the reflection the range [0, 2π[has to be considered in a circular way, therefore the reflected Y_i is still defined for the same points X as before.

Finally, given the probability functions Y_1 ... Y_C defined for the points X and considering equal mixing proportions, the final Gaussian and Gamma mixture distribution is obtained as the average of the components' probability functions.

Louvain algorithm for community detection

The Louvain algorithm [44] is separated into two phases, which are repeated iteratively.

At first every node in the (weighted) network represents a community in itself. In the first phase, for each node i, it considers its neighbours j and evaluates the gain in modularity that would take place by removing i from its community and placing it in the community of j. The node i is then placed in the community j for which this gain is maximum, but only if the gain is positive. If no gain is possible node i stays in its original community. This process is applied until no further improvement can be achieved.

In the second phase the algorithm builds a new network whose nodes are the communities found in the first phase, whereas the weights of the links between the new nodes are given by the sum of the weight of the links between nodes in the corresponding two communities. Links between nodes of the same community lead to self-loops for this community in the new network.

Once the new network has been built, the two phase process is iterated until there are no more changes and a maximum of modularity has been obtained. The number of iterations determines the height of the hierarchy of communities detected by the algorithm.

For each hierarchical level there is a possible partition to compare to the ground truth annotation. In this case, the hierarchical level considered is the one that guarantees the best match, therefore the detected partition that gives the highest NMI value.

We used the R function *multilevel.community*, an implementation of the method available in the *igraph* package [96].

Normalized Mutual Information

The evaluation of the community detection has been performed using the Normalized Mutual Information (NMI) as in [98]. The entropy can be defined as the information contained in a distribution *p(x)* in the following way:

$$H(X) = \sum_{x \in X} p(x) \log p(x)$$

The mutual information is the shared information between two distributions:

$$I(X,Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p_1(x)p_2(y)}\right)$$

To normalize the value between 0 and 1 the following formula can be applied:

$$NMI = \frac{I(X,Y)}{\sqrt{H(X)H(Y)}}$$

If we consider a partition of the nodes in communities as a distribution (probability of one node falling into one community), we can compute the matching between the annotations obtained by the community detection algorithm and the ground truth communities of a network as follows:

$$H(C_D) = \sum_{h=1}^{n_D} \frac{n_h^D}{N} \log(\frac{n_h^D}{N})$$

$$H(C_T) = \sum_{l=1}^{n_T} \frac{n_l^T}{N} \log(\frac{n_l^T}{N})$$

$$I(C_D, C_T) = \sum_h \sum_l \frac{n_{h,l}}{N} \log\left(\frac{n_{h,l}}{n_h^D n_l^T}\right).$$

$$NMI(C_D, C_T) = \frac{I(C_D, C_T)}{\sqrt{H(C_D)H(C_T)}}$$

Where:

N - number of nodes;

n^D, n^T - number of communities detected by the algorithm (*D*) or ground truth (*T*);

$n_{h,l}$ - number of nodes assigned to the h-th community by the algorithm and to the l-th community according to the ground truth annotation.

We used the MATLAB implementation available at http://commdetect.weebly.com. As suggested in the code, when $\frac{N}{n^T} \leq 100$, the NMI should be adjusted in order to correct for chance [99].

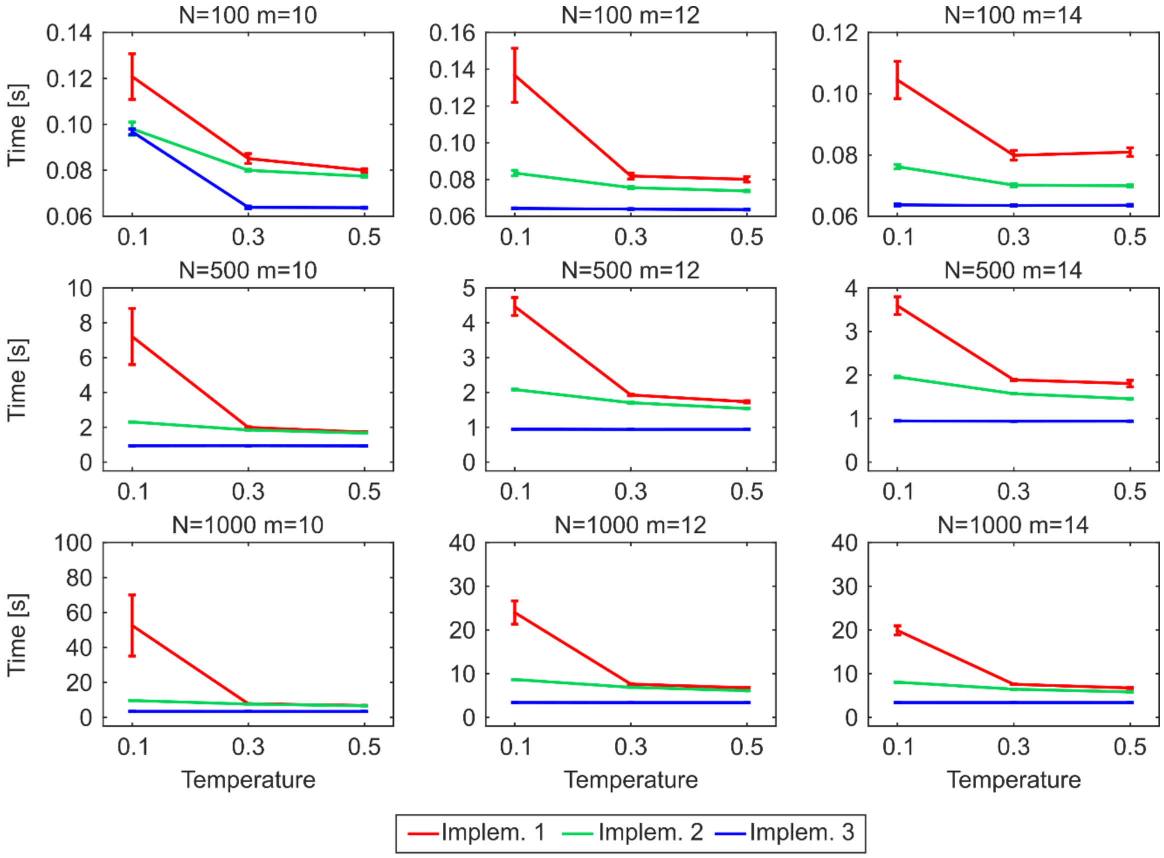


Figure 1. Time performance for generating PSO networks.
Synthetic networks have been generated using the PSO model with parameters N = [100, 500, 1000], m = [10, 12, 14], T = [0.1, 0.3, 0.5] and γ = 3. For each combination of parameters, 10 networks have been generated using the 3 different implementations. The plots report for each parameter combination the mean computational time and standard error over the random repetitions.

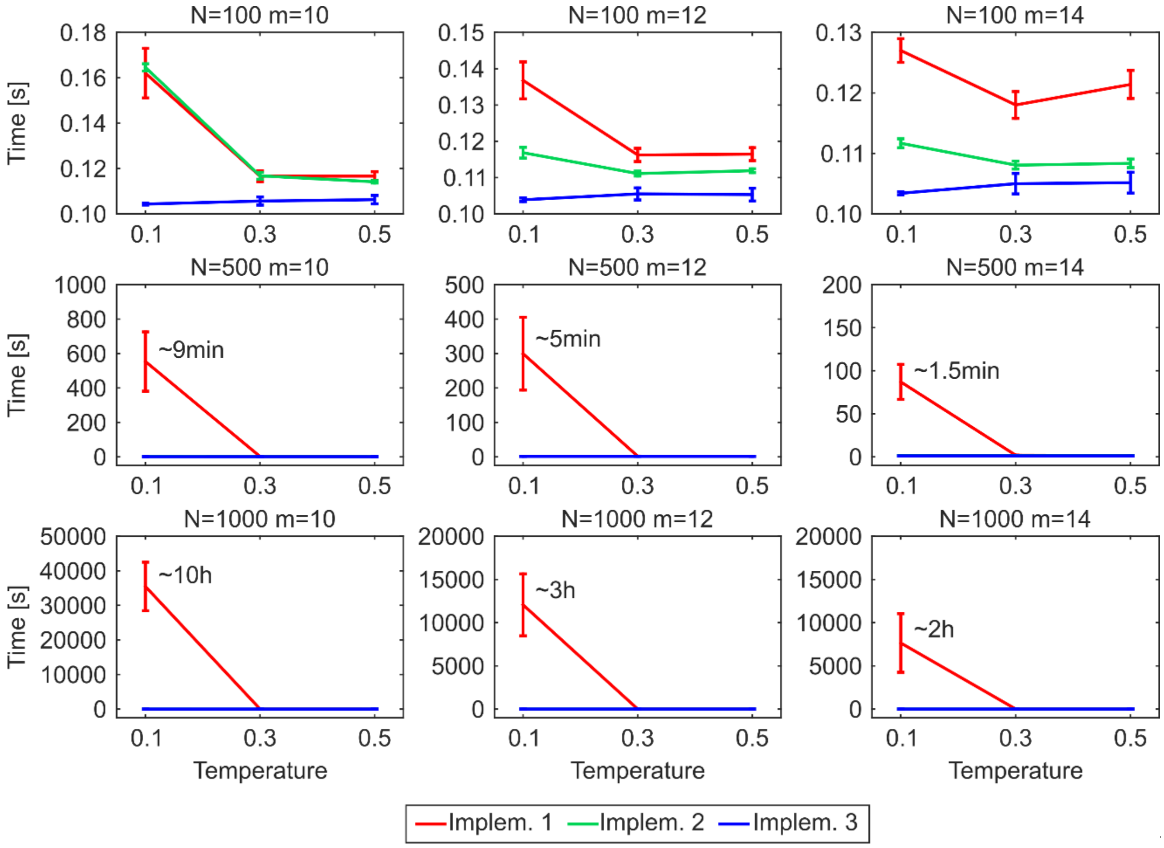


Figure 2. Time performance for generating nPSO networks with 4 communities.
Synthetic networks have been generated using the nPSO model with parameters $N = [100, 500, 1000]$, $m = [10, 12, 14]$, $T = [0.1, 0.3, 0.5]$, $\gamma = 3$ and angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components $C = 4$. For each combination of parameters, 10 networks have been generated using the 3 different implementations. The plots report for each parameter combination the mean computational time and standard error over the random repetitions.

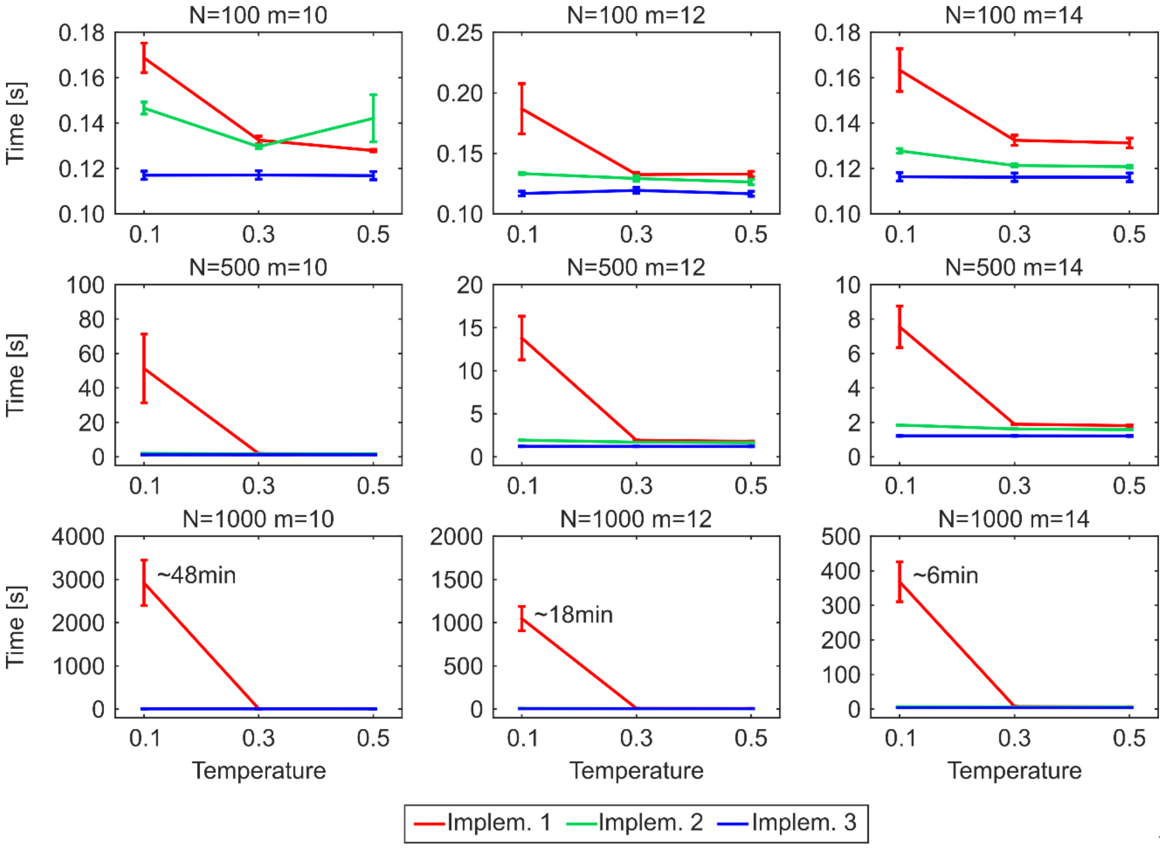


Figure 3. Time performance for generating nPSO networks with 8 communities.
Synthetic networks have been generated using the nPSO model with parameters $N = [100, 500, 1000]$, $m = [10, 12, 14]$, $T = [0.1, 0.3, 0.5]$, $\gamma = 3$ and angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components $C = 8$. For each combination of parameters, 10 networks have been generated using the 3 different implementations. The plots report for each parameter combination the mean computational time and standard error over the random repetitions.

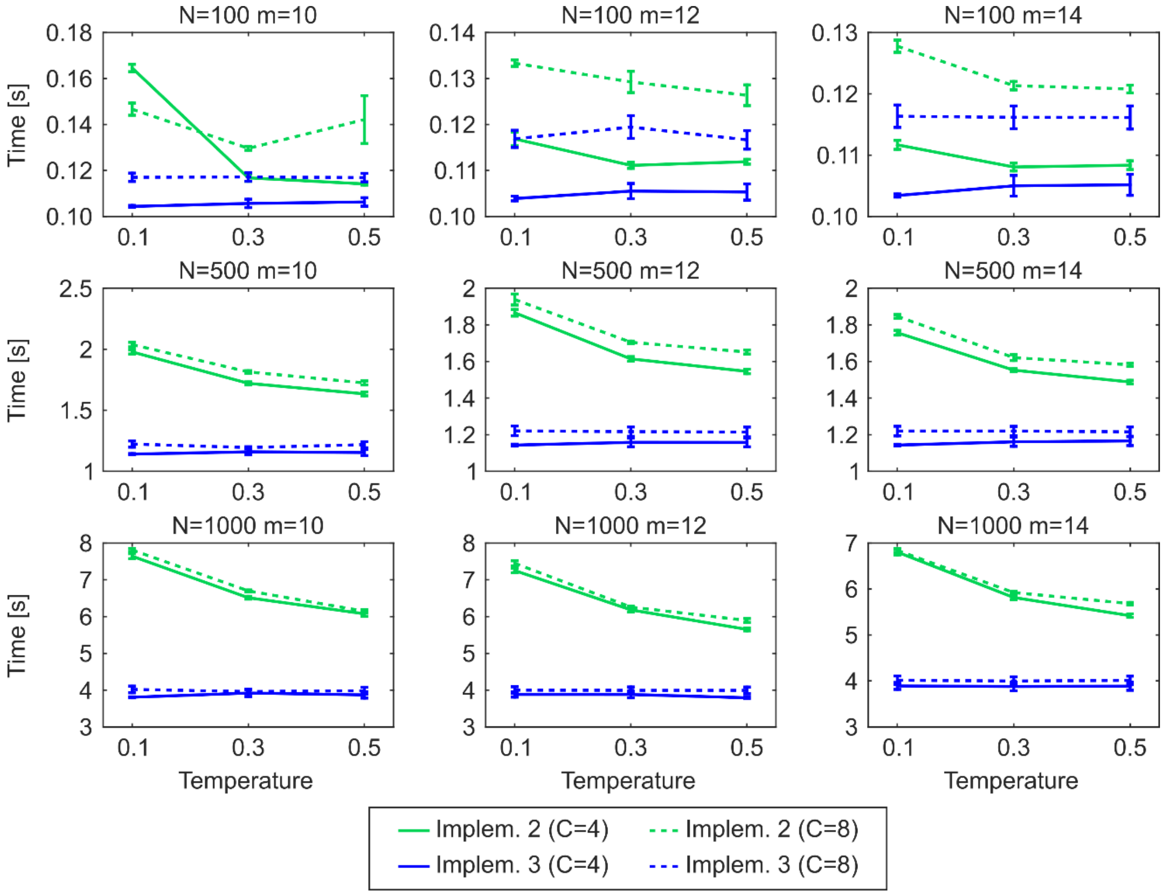


Figure 4. Time performance for generating nPSO networks: implementations 2-3.
Due to the different scale of the computational time of implementation 1, the figure reports a more detailed comparison of the time performance for generating nPSO networks with 4 and 8 communities only for implementations 2 and 3.

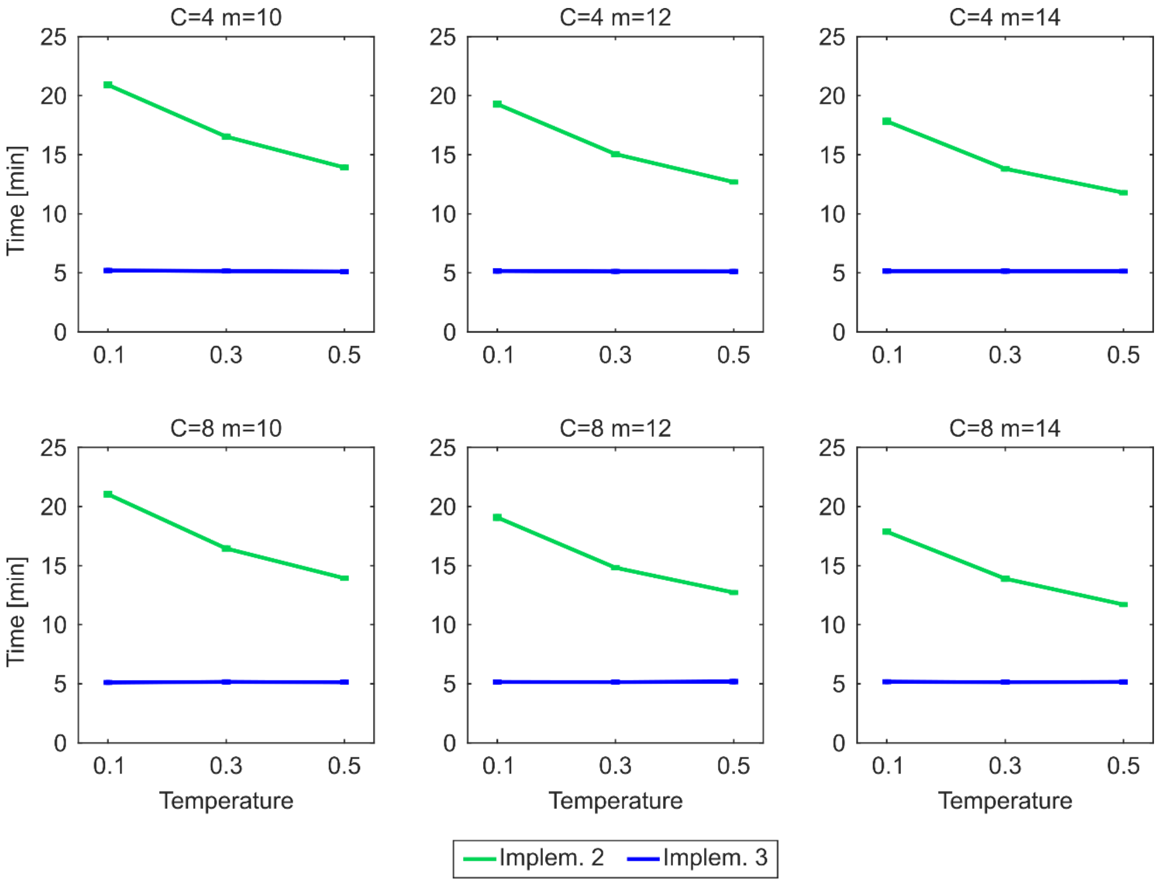


Figure 5. Time performance for generating large-size nPSO networks.
Synthetic networks have been generated using the nPSO model with parameters $N = 10000$, $m = [10, 12, 14]$, $T = [0.1, 0.3, 0.5]$, $\gamma = 3$ and angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components $C = [4, 8]$. For each combination of parameters, 10 networks have been generated using the implementations 2 and 3. The plots report for each parameter combination the mean computational time and standard error over the random repetitions.

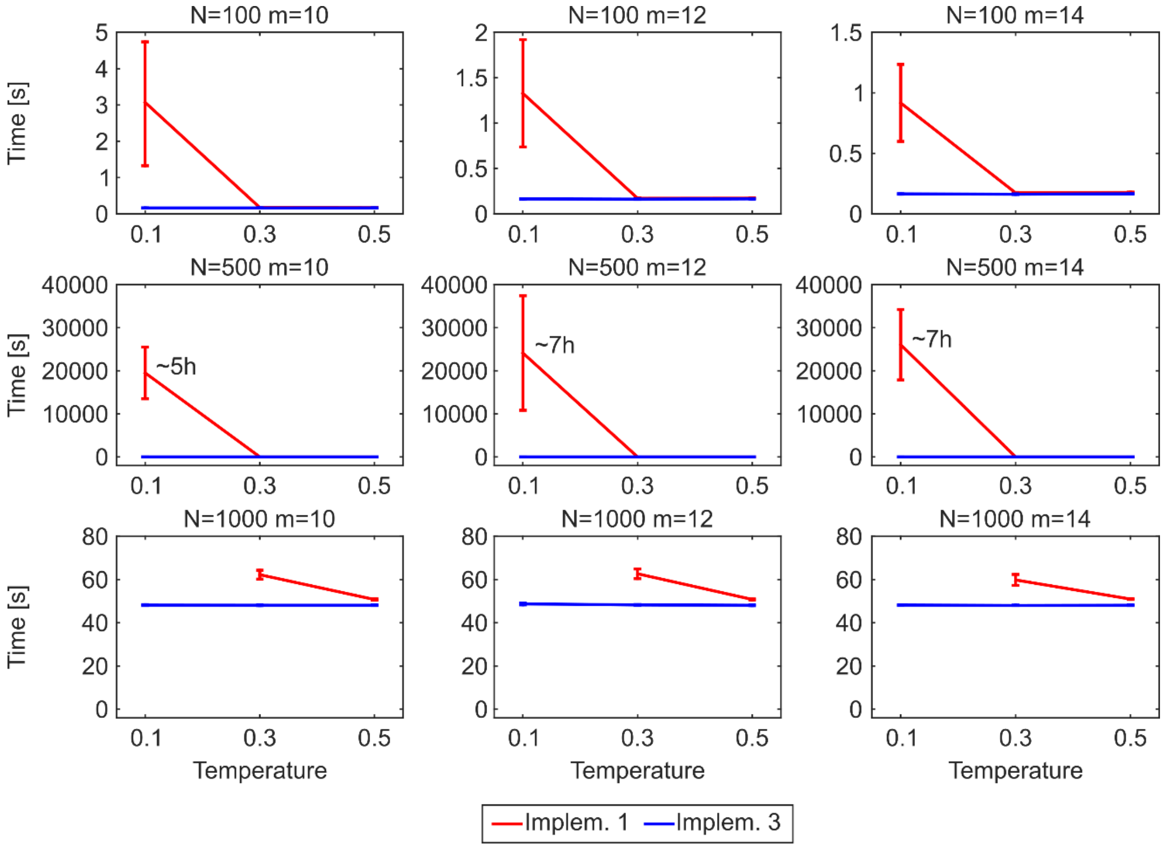


Figure 6. Time performance for generating GPA networks with Λ = 0.1.

Synthetic networks have been generated using the GPA model with parameters N = [100, 500, 1000], m = [10, 12, 14], T = [0.1, 0.3, 0.5], γ = 3 and Λ = 0.1. For each combination of parameters, 10 networks have been generated using the implementations 1 and 3. The plots report for each parameter combination the mean computational time and standard error over the random repetitions. Note that for T = 0.1 and N = 1000 the results of implementation 1 are not reported since the computational time required to generate the networks was too high (more than one month).

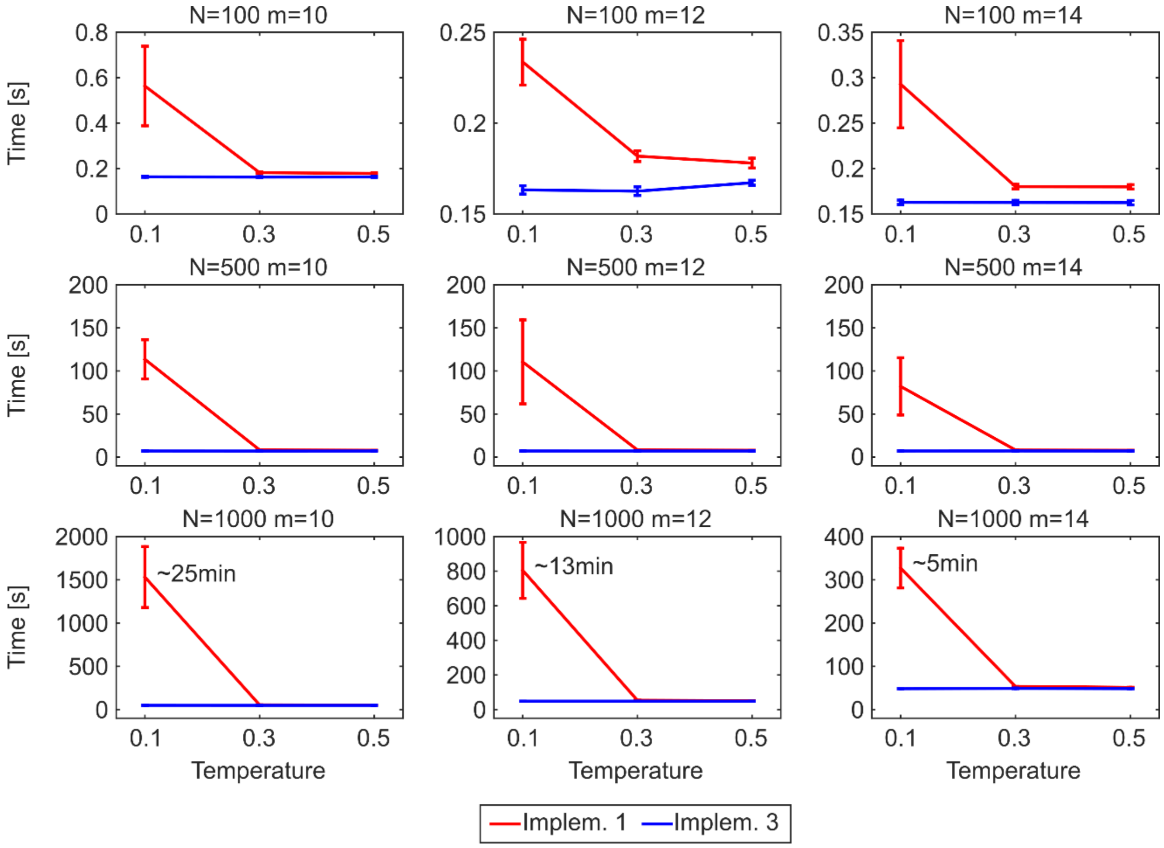


Figure 7. Time performance for generating GPA networks with $\Lambda = 1$.
Synthetic networks have been generated using the GPA model with parameters $N = [100, 500, 1000]$, $m = [10, 12, 14]$, $T = [0.1, 0.3, 0.5]$, $\gamma = 3$ and $\Lambda = 1$. For each combination of parameters, 10 networks have been generated using the implementations 1 and 3. The plots report for each parameter combination the mean computational time and standard error over the random repetitions.

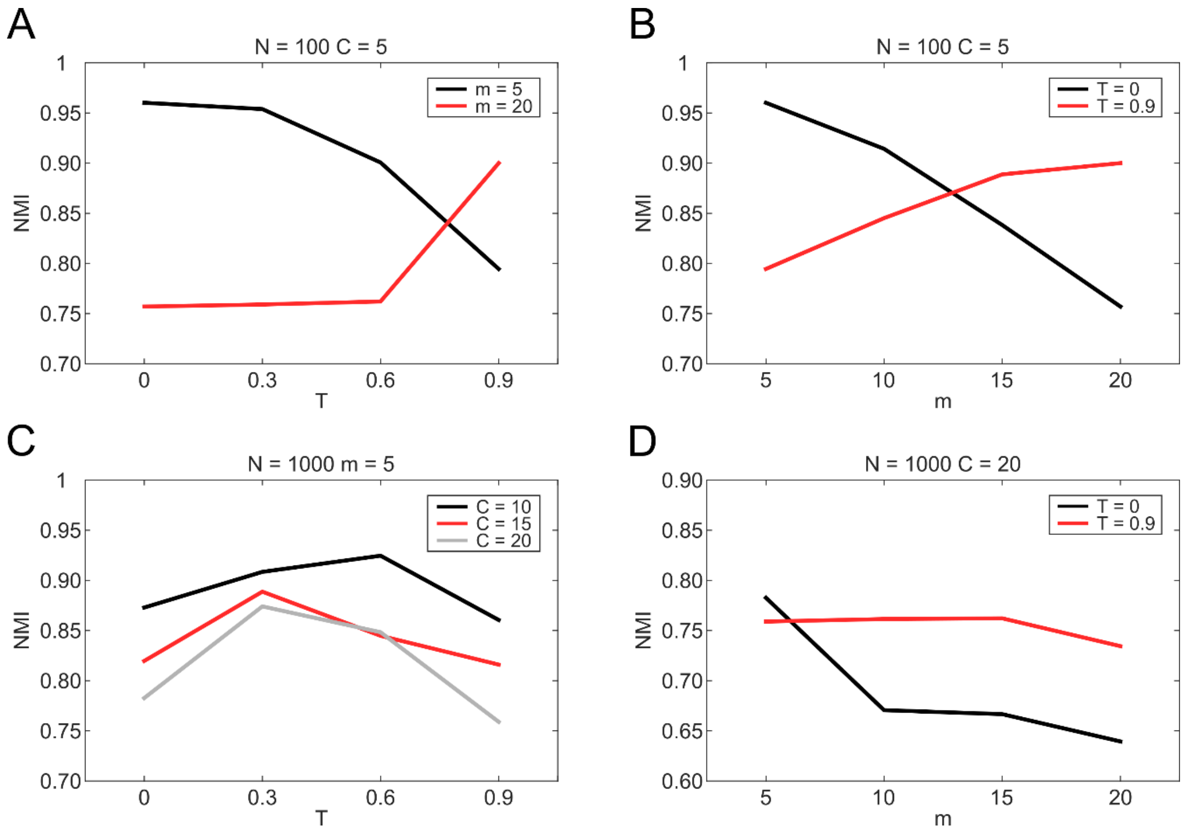
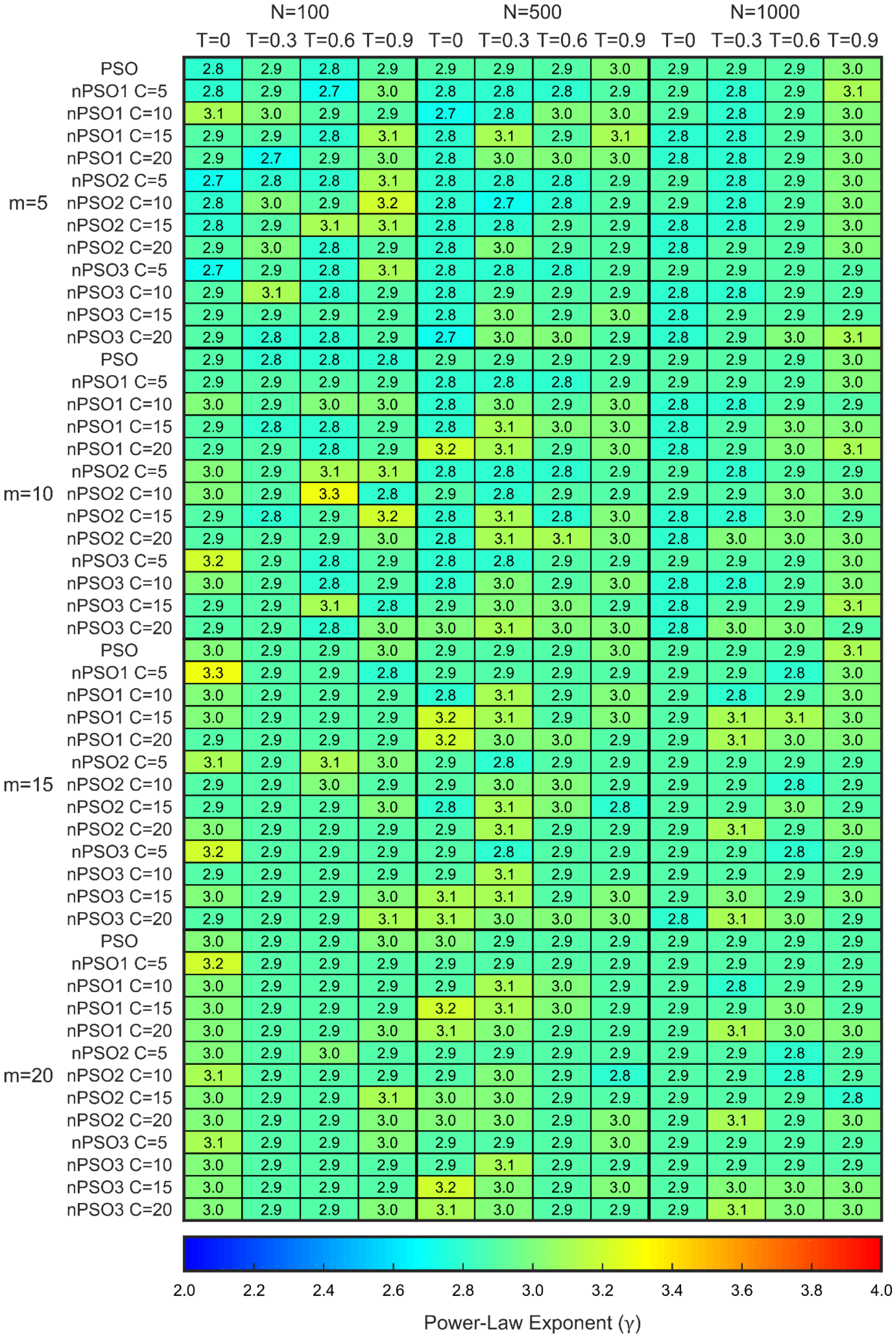


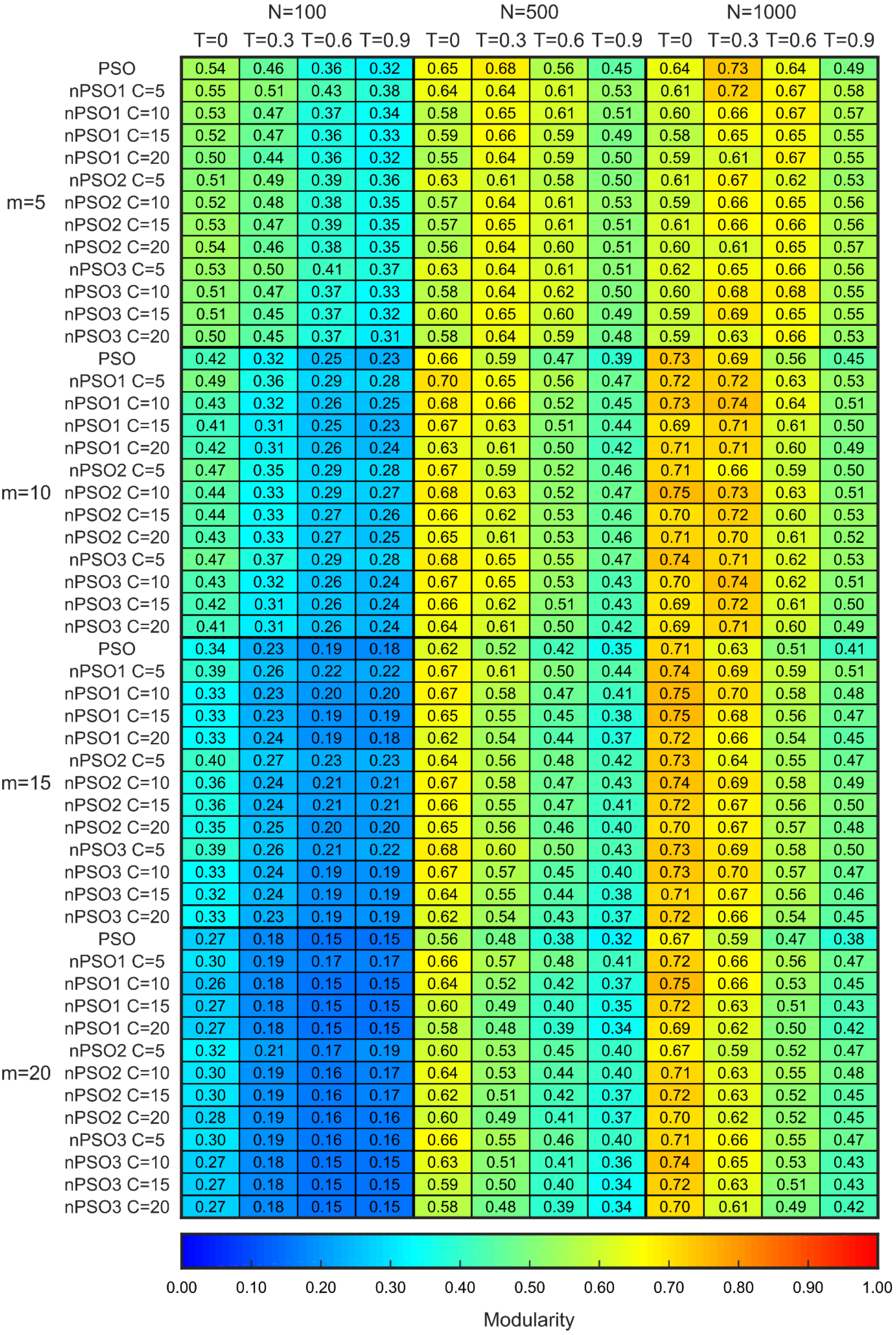
Figure 8. Detectability of the nPSO communities: particular cases.
The figure highlights some particular parameter combinations from the Louvain-NMI heatmap in Figure 4 of the main article. (A) $N = 100$, $m = [5, 20]$, $T = [0, 0.3, 0.6, 0.9]$, $\gamma = 3$, $C = 5$ (nPSO1). (B) $N = 100$, $m = [5, 10, 15, 20]$, $T = [0, 0.9]$, $\gamma = 3$, $C = 5$ (nPSO1). (C) $N = 1000$, $m = 5$, $T = [0, 0.3, 0.6, 0.9]$, $\gamma = 3$, $C = [10, 15, 20]$ (nPSO1). (D) $N = 1000$, $m = [5, 10, 15, 20]$, $T = [0, 0.9]$, $\gamma = 3$, $C = 20$ (nPSO1).



		N=100				N=500				N=1000			
		T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9
m=5	PSO	2.8	2.9	2.8	2.9	2.9	2.9	2.9	3.0	2.9	2.9	2.9	3.0
	nPSO1 C=5	2.8	2.9	2.7	3.0	2.8	2.8	2.8	2.9	2.9	2.8	2.9	3.1
	nPSO1 C=10	3.1	3.0	2.9	2.9	2.7	2.8	3.0	3.0	2.9	2.8	2.9	3.0
	nPSO1 C=15	2.9	2.9	2.8	3.1	2.8	3.1	2.9	3.1	2.8	2.8	2.9	3.0
	nPSO1 C=20	2.9	2.7	2.9	3.0	2.8	3.0	3.0	3.0	2.8	2.8	2.9	3.0
	nPSO2 C=5	2.7	2.8	2.8	3.1	2.8	2.8	2.8	2.9	2.9	2.8	2.9	3.0
	nPSO2 C=10	2.8	3.0	2.9	3.2	2.8	2.7	2.8	2.9	2.9	2.8	2.9	3.0
	nPSO2 C=15	2.8	2.9	3.1	3.1	2.8	2.8	2.9	2.9	2.8	2.8	2.9	3.0
	nPSO2 C=20	2.9	3.0	2.8	2.9	2.8	3.0	2.9	2.9	2.8	2.9	2.9	3.0
	nPSO3 C=5	2.7	2.9	2.8	3.1	2.8	2.8	2.8	2.9	2.9	2.9	2.9	2.9
	nPSO3 C=10	2.9	3.1	2.8	2.9	2.8	2.9	2.9	2.9	2.8	2.8	2.9	2.9
	nPSO3 C=15	2.9	2.9	2.9	2.9	2.8	3.0	2.9	3.0	2.8	2.9	2.9	2.9
	nPSO3 C=20	2.9	2.8	2.8	2.9	2.7	3.0	3.0	2.9	2.8	2.9	3.0	3.1
m=10	PSO	2.9	2.8	2.8	2.8	2.9	2.9	2.9	2.9	2.9	2.9	2.9	3.0
	nPSO1 C=5	2.9	2.9	2.9	2.9	2.8	2.8	2.8	2.9	2.9	2.9	2.9	3.0
	nPSO1 C=10	3.0	2.9	3.0	3.0	2.8	3.0	2.9	3.0	2.8	2.8	2.9	2.9
	nPSO1 C=15	2.9	2.8	2.8	2.9	2.8	3.1	3.0	3.0	2.8	2.9	3.0	3.0
	nPSO1 C=20	2.9	2.9	2.8	2.9	3.2	3.1	2.9	3.0	2.8	2.9	3.0	3.1
	nPSO2 C=5	3.0	2.9	3.1	3.1	2.8	2.8	2.8	2.9	2.9	2.8	2.9	2.9
	nPSO2 C=10	3.0	2.9	3.3	2.8	2.9	2.8	2.9	2.9	2.9	2.9	3.0	3.0
	nPSO2 C=15	2.9	2.8	2.9	3.2	2.8	3.1	2.8	3.0	2.8	2.8	3.0	2.9
	nPSO2 C=20	2.9	2.9	2.9	3.0	2.8	3.1	3.1	3.0	2.8	3.0	3.0	3.0
	nPSO3 C=5	3.2	2.9	2.8	2.9	2.8	2.8	2.9	2.9	2.9	2.9	2.9	3.0
	nPSO3 C=10	3.0	2.9	2.8	2.9	2.8	3.0	2.9	3.0	2.8	2.8	2.9	3.0
	nPSO3 C=15	2.9	2.9	3.1	2.8	2.9	3.0	3.0	2.9	2.8	2.9	2.9	3.1
	nPSO3 C=20	2.9	2.9	2.8	3.0	3.0	3.1	3.0	3.0	2.8	3.0	3.0	2.9
m=15	PSO	3.0	2.9	2.9	3.0	2.9	2.9	2.9	3.0	2.9	2.9	2.9	3.1
	nPSO1 C=5	3.3	2.9	2.9	2.8	2.9	2.9	2.9	2.9	2.9	2.9	2.8	3.0
	nPSO1 C=10	3.0	2.9	2.9	2.9	2.8	3.1	2.9	3.0	2.9	2.8	2.9	3.0
	nPSO1 C=15	3.0	2.9	2.9	2.9	3.2	3.1	2.9	3.0	2.9	3.1	3.1	3.0
	nPSO1 C=20	2.9	2.9	2.9	2.9	3.2	3.0	3.0	2.9	2.9	3.1	3.0	3.0
	nPSO2 C=5	3.1	2.9	3.1	3.0	2.9	2.8	2.9	2.9	2.9	2.9	2.9	2.9
	nPSO2 C=10	2.9	2.9	3.0	2.9	2.9	3.0	3.0	2.9	2.9	2.9	2.8	2.9
	nPSO2 C=15	2.9	2.9	2.9	3.0	2.8	3.1	3.0	2.8	2.9	2.9	3.0	2.9
	nPSO2 C=20	3.0	2.9	2.9	2.9	2.9	3.1	2.9	2.9	2.9	3.1	2.9	3.0
	nPSO3 C=5	3.2	2.9	2.9	2.9	2.9	2.8	2.9	2.9	2.9	2.9	2.8	2.9
	nPSO3 C=10	2.9	2.9	2.9	2.9	2.9	3.1	2.9	2.9	2.9	2.9	2.9	2.9
	nPSO3 C=15	3.0	2.9	2.9	3.0	3.1	3.1	2.9	3.0	2.9	3.0	2.9	3.0
	nPSO3 C=20	2.9	2.9	2.9	3.1	3.1	3.0	3.0	3.0	2.8	3.1	3.0	2.9
m=20	PSO	3.0	2.9	2.9	3.0	3.0	2.9	2.9	2.9	2.9	2.9	2.9	2.9
	nPSO1 C=5	3.2	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9
	nPSO1 C=10	3.0	2.9	2.9	2.9	2.9	3.1	3.0	2.9	2.9	2.8	2.9	2.9
	nPSO1 C=15	3.0	2.9	2.9	2.9	3.2	3.1	3.0	2.9	2.9	2.9	3.0	2.9
	nPSO1 C=20	3.0	2.9	2.9	3.0	3.1	3.0	2.9	2.9	2.9	3.1	3.0	3.0
	nPSO2 C=5	3.0	2.9	3.0	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.8	2.9
	nPSO2 C=10	3.1	2.9	2.9	2.9	2.9	3.0	2.9	2.8	2.9	2.9	2.8	2.9
	nPSO2 C=15	3.0	2.9	2.9	3.1	3.0	3.0	2.9	2.9	2.9	2.9	2.9	2.8
	nPSO2 C=20	3.0	2.9	2.9	3.0	3.0	3.0	2.9	3.0	2.9	3.1	2.9	3.0
	nPSO3 C=5	3.1	2.9	2.9	3.0	2.9	2.9	2.9	3.0	2.9	2.9	2.9	2.9
	nPSO3 C=10	3.0	2.9	2.9	2.9	2.9	3.1	2.9	2.9	2.9	2.9	2.9	2.9
	nPSO3 C=15	3.0	2.9	2.9	2.9	3.2	3.0	2.9	3.0	2.9	3.0	3.0	3.0
	nPSO3 C=20	3.0	2.9	2.9	3.0	3.1	3.0	2.9	2.9	2.9	3.1	3.0	3.0

Figure 9. Power-law exponent (γ) of the PSO and nPSO networks.

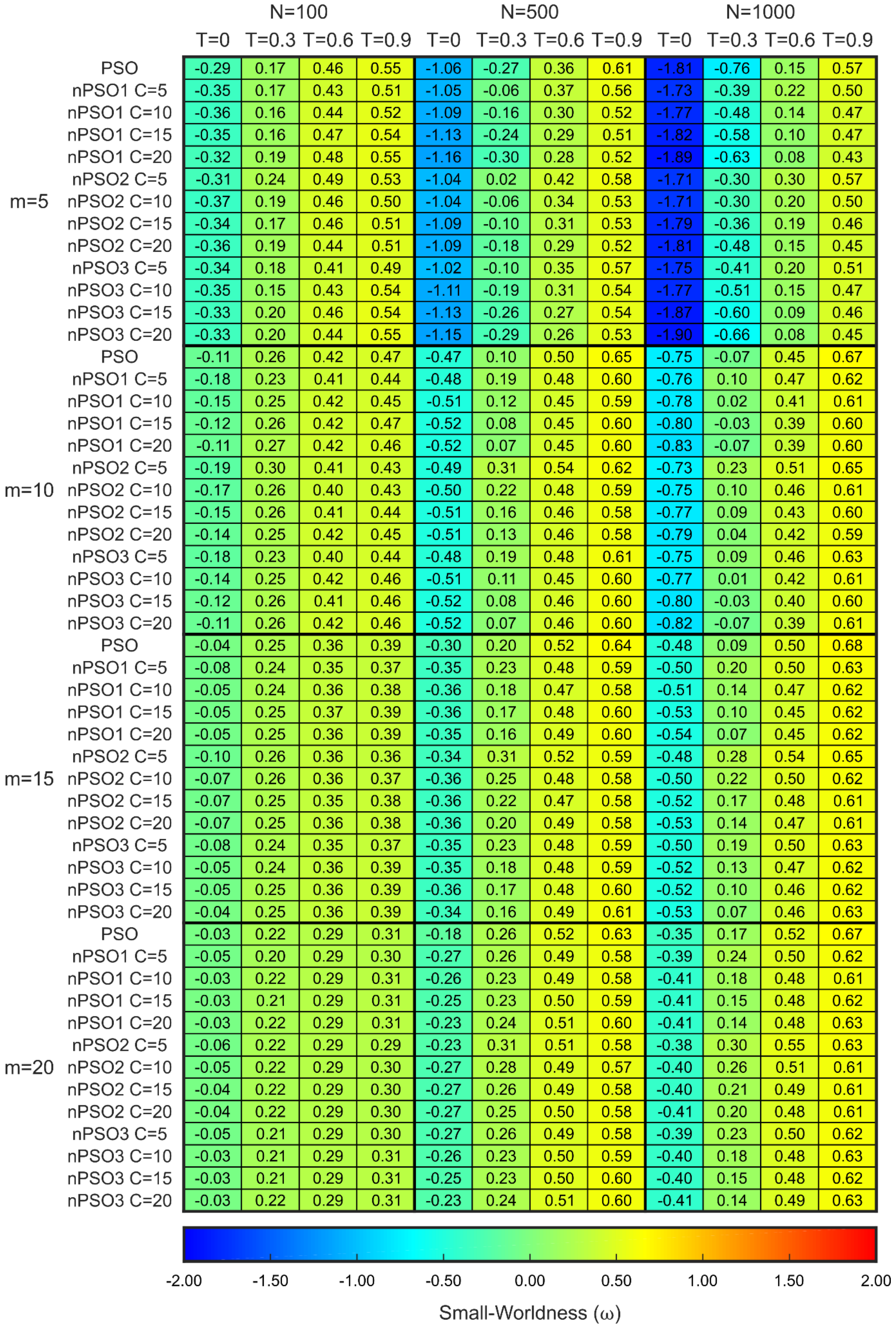
Synthetic networks have been generated using the nPSO model with parameters N = [100, 500, 1000], m = [5, 10, 15, 20], T = [0, 0.3, 0.6, 0.9], γ = 3 and angular coordinates sampled according to mixture distributions of three different kinds with components C = [5, 10, 15, 20]: Gaussian mixture with equal proportions (nPSO1), Gaussian mixture with random proportions (nPSO2), Gaussian and Gamma mixture with equal proportions (nPSO3). For more details on the parameters of the mixture distributions please refer to the Methods. Furthermore, synthetic networks have been generated also using the PSO model with the same parameters N, m, T and γ. For each combination of parameters, 10 networks have been generated and the power-law exponent (γ) has been computed. The heatmap reports for each parameter combination the mean power-law exponent (γ), coloured according to a blue-to-red colormap in the range [2, 4].



		N=100				N=500				N=1000			
		T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9
m=5	PSO	0.54	0.46	0.36	0.32	0.65	0.68	0.56	0.45	0.64	0.73	0.64	0.49
	nPSO1 C=5	0.55	0.51	0.43	0.38	0.64	0.64	0.61	0.53	0.61	0.72	0.67	0.58
	nPSO1 C=10	0.53	0.47	0.37	0.34	0.58	0.65	0.61	0.51	0.60	0.66	0.67	0.57
	nPSO1 C=15	0.52	0.47	0.36	0.33	0.59	0.66	0.59	0.49	0.58	0.65	0.65	0.55
	nPSO1 C=20	0.50	0.44	0.36	0.32	0.55	0.64	0.59	0.50	0.59	0.61	0.67	0.55
	nPSO2 C=5	0.51	0.49	0.39	0.36	0.63	0.61	0.58	0.50	0.61	0.67	0.62	0.53
	nPSO2 C=10	0.52	0.48	0.38	0.35	0.57	0.64	0.61	0.53	0.59	0.66	0.65	0.56
	nPSO2 C=15	0.53	0.47	0.39	0.35	0.57	0.65	0.61	0.51	0.61	0.66	0.66	0.56
	nPSO2 C=20	0.54	0.46	0.38	0.35	0.56	0.64	0.60	0.51	0.60	0.61	0.65	0.57
	nPSO3 C=5	0.53	0.50	0.41	0.37	0.63	0.64	0.61	0.51	0.62	0.65	0.66	0.56
	nPSO3 C=10	0.51	0.47	0.37	0.33	0.58	0.64	0.62	0.50	0.60	0.68	0.68	0.55
	nPSO3 C=15	0.51	0.45	0.37	0.32	0.60	0.65	0.60	0.49	0.59	0.69	0.65	0.55
	nPSO3 C=20	0.50	0.45	0.37	0.31	0.58	0.64	0.59	0.48	0.59	0.63	0.66	0.53
m=10	PSO	0.42	0.32	0.25	0.23	0.66	0.59	0.47	0.39	0.73	0.69	0.56	0.45
	nPSO1 C=5	0.49	0.36	0.29	0.28	0.70	0.65	0.56	0.47	0.72	0.72	0.63	0.53
	nPSO1 C=10	0.43	0.32	0.26	0.25	0.68	0.66	0.52	0.45	0.73	0.74	0.64	0.51
	nPSO1 C=15	0.41	0.31	0.25	0.23	0.67	0.63	0.51	0.44	0.69	0.71	0.61	0.50
	nPSO1 C=20	0.42	0.31	0.26	0.24	0.63	0.61	0.50	0.42	0.71	0.71	0.60	0.49
	nPSO2 C=5	0.47	0.35	0.29	0.28	0.67	0.59	0.52	0.46	0.71	0.66	0.59	0.50
	nPSO2 C=10	0.44	0.33	0.29	0.27	0.68	0.63	0.52	0.47	0.75	0.73	0.63	0.51
	nPSO2 C=15	0.44	0.33	0.27	0.26	0.66	0.62	0.53	0.46	0.70	0.72	0.60	0.53
	nPSO2 C=20	0.43	0.33	0.27	0.25	0.65	0.61	0.53	0.46	0.71	0.70	0.61	0.52
	nPSO3 C=5	0.47	0.37	0.29	0.28	0.68	0.65	0.55	0.47	0.74	0.71	0.62	0.53
	nPSO3 C=10	0.43	0.32	0.26	0.24	0.67	0.65	0.53	0.43	0.70	0.74	0.62	0.51
	nPSO3 C=15	0.42	0.31	0.26	0.24	0.66	0.62	0.51	0.43	0.69	0.72	0.61	0.50
	nPSO3 C=20	0.41	0.31	0.26	0.24	0.64	0.61	0.50	0.42	0.69	0.71	0.60	0.49
m=15	PSO	0.34	0.23	0.19	0.18	0.62	0.52	0.42	0.35	0.71	0.63	0.51	0.41
	nPSO1 C=5	0.39	0.26	0.22	0.22	0.67	0.61	0.50	0.44	0.74	0.69	0.59	0.51
	nPSO1 C=10	0.33	0.23	0.20	0.20	0.67	0.58	0.47	0.41	0.75	0.70	0.58	0.48
	nPSO1 C=15	0.33	0.23	0.19	0.19	0.65	0.55	0.45	0.38	0.75	0.68	0.56	0.47
	nPSO1 C=20	0.33	0.24	0.19	0.18	0.62	0.54	0.44	0.37	0.72	0.66	0.54	0.45
	nPSO2 C=5	0.40	0.27	0.23	0.23	0.64	0.56	0.48	0.42	0.73	0.64	0.55	0.47
	nPSO2 C=10	0.36	0.24	0.21	0.21	0.67	0.58	0.47	0.43	0.74	0.69	0.58	0.49
	nPSO2 C=15	0.36	0.24	0.21	0.21	0.66	0.55	0.47	0.41	0.72	0.67	0.56	0.50
	nPSO2 C=20	0.35	0.25	0.20	0.20	0.65	0.56	0.46	0.40	0.70	0.67	0.57	0.48
	nPSO3 C=5	0.39	0.26	0.21	0.22	0.68	0.60	0.50	0.43	0.73	0.69	0.58	0.50
	nPSO3 C=10	0.33	0.24	0.19	0.19	0.67	0.57	0.45	0.40	0.73	0.70	0.57	0.47
	nPSO3 C=15	0.32	0.24	0.19	0.19	0.64	0.55	0.44	0.38	0.71	0.67	0.56	0.46
	nPSO3 C=20	0.33	0.23	0.19	0.19	0.62	0.54	0.43	0.37	0.72	0.66	0.54	0.45
m=20	PSO	0.27	0.18	0.15	0.15	0.56	0.48	0.38	0.32	0.67	0.59	0.47	0.38
	nPSO1 C=5	0.30	0.19	0.17	0.17	0.66	0.57	0.48	0.41	0.72	0.66	0.56	0.47
	nPSO1 C=10	0.26	0.18	0.15	0.15	0.64	0.52	0.42	0.37	0.75	0.66	0.53	0.45
	nPSO1 C=15	0.27	0.18	0.15	0.15	0.60	0.49	0.40	0.35	0.72	0.63	0.51	0.43
	nPSO1 C=20	0.27	0.18	0.15	0.15	0.58	0.48	0.39	0.34	0.69	0.62	0.50	0.42
	nPSO2 C=5	0.32	0.21	0.17	0.19	0.60	0.53	0.45	0.40	0.67	0.59	0.52	0.47
	nPSO2 C=10	0.30	0.19	0.16	0.17	0.64	0.53	0.44	0.40	0.71	0.63	0.55	0.48
	nPSO2 C=15	0.30	0.19	0.16	0.17	0.62	0.51	0.42	0.37	0.72	0.63	0.52	0.45
	nPSO2 C=20	0.28	0.19	0.16	0.16	0.60	0.49	0.41	0.37	0.70	0.62	0.52	0.45
	nPSO3 C=5	0.30	0.19	0.16	0.16	0.66	0.55	0.46	0.40	0.71	0.66	0.55	0.47
	nPSO3 C=10	0.27	0.18	0.15	0.15	0.63	0.51	0.41	0.36	0.74	0.65	0.53	0.43
	nPSO3 C=15	0.27	0.18	0.15	0.15	0.59	0.50	0.40	0.34	0.72	0.63	0.51	0.43
	nPSO3 C=20	0.27	0.18	0.15	0.15	0.58	0.48	0.39	0.34	0.70	0.61	0.49	0.42

Figure 10. Modularity of the PSO and nPSO networks.

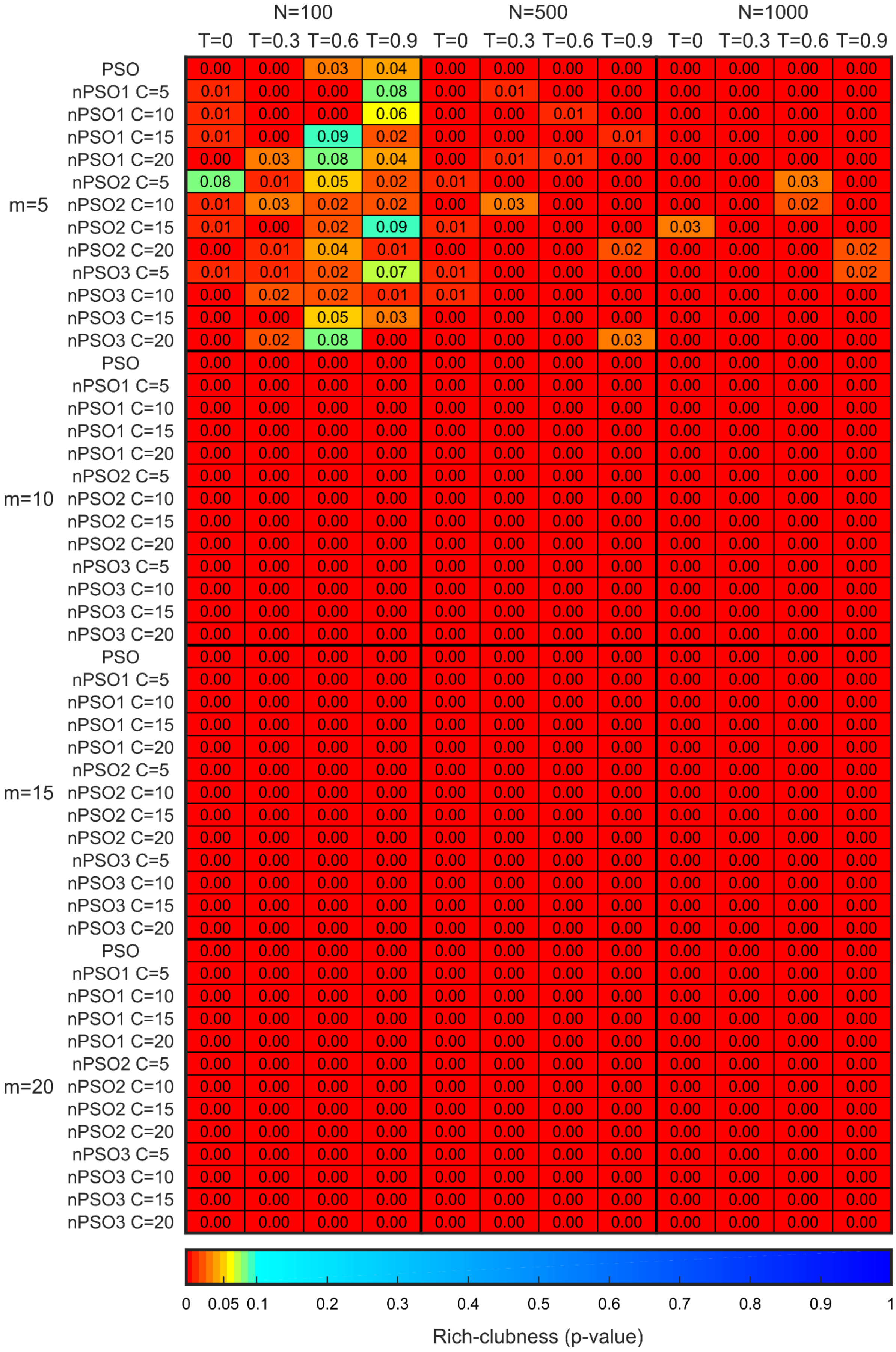
Synthetic networks have been generated using the nPSO model with parameters N = [100, 500, 1000], m = [5, 10, 15, 20], T = [0, 0.3, 0.6, 0.9], γ = 3 and angular coordinates sampled according to mixture distributions of three different kinds with components C = [5, 10, 15, 20]: Gaussian mixture with equal proportions (nPSO1), Gaussian mixture with random proportions (nPSO2), Gaussian and Gamma mixture with equal proportions (nPSO3). For more details on the parameters of the mixture distributions please refer to the Methods. Furthermore, synthetic networks have been generated also using the PSO model with the same parameters N, m, T and γ. For each combination of parameters, 10 networks have been generated and the modularity has been computed. The heatmap reports for each parameter combination the mean modularity, coloured according to a blue-to-red colormap in the range [0, 1].



		N=100				N=500				N=1000			
		T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9
m=5	PSO	-0.29	0.17	0.46	0.55	-1.06	-0.27	0.36	0.61	-1.81	-0.76	0.15	0.57
	nPSO1 C=5	-0.35	0.17	0.43	0.51	-1.05	-0.06	0.37	0.56	-1.73	-0.39	0.22	0.50
	nPSO1 C=10	-0.36	0.16	0.44	0.52	-1.09	-0.16	0.30	0.52	-1.77	-0.48	0.14	0.47
	nPSO1 C=15	-0.35	0.16	0.47	0.54	-1.13	-0.24	0.29	0.51	-1.82	-0.58	0.10	0.47
	nPSO1 C=20	-0.32	0.19	0.48	0.55	-1.16	-0.30	0.28	0.52	-1.89	-0.63	0.08	0.43
	nPSO2 C=5	-0.31	0.24	0.49	0.53	-1.04	0.02	0.42	0.58	-1.71	-0.30	0.30	0.57
	nPSO2 C=10	-0.37	0.19	0.46	0.50	-1.04	-0.06	0.34	0.53	-1.71	-0.30	0.20	0.50
	nPSO2 C=15	-0.34	0.17	0.46	0.51	-1.09	-0.10	0.31	0.53	-1.79	-0.36	0.19	0.46
	nPSO2 C=20	-0.36	0.19	0.44	0.51	-1.09	-0.18	0.29	0.52	-1.81	-0.48	0.15	0.45
	nPSO3 C=5	-0.34	0.18	0.41	0.49	-1.02	-0.10	0.35	0.57	-1.75	-0.41	0.20	0.51
	nPSO3 C=10	-0.35	0.15	0.43	0.54	-1.11	-0.19	0.31	0.54	-1.77	-0.51	0.15	0.47
	nPSO3 C=15	-0.33	0.20	0.46	0.54	-1.13	-0.26	0.27	0.54	-1.87	-0.60	0.09	0.46
	nPSO3 C=20	-0.33	0.20	0.44	0.55	-1.15	-0.29	0.26	0.53	-1.90	-0.66	0.08	0.45
m=10	PSO	-0.11	0.26	0.42	0.47	-0.47	0.10	0.50	0.65	-0.75	-0.07	0.45	0.67
	nPSO1 C=5	-0.18	0.23	0.41	0.44	-0.48	0.19	0.48	0.60	-0.76	0.10	0.47	0.62
	nPSO1 C=10	-0.15	0.25	0.42	0.45	-0.51	0.12	0.45	0.59	-0.78	0.02	0.41	0.61
	nPSO1 C=15	-0.12	0.26	0.42	0.47	-0.52	0.08	0.45	0.60	-0.80	-0.03	0.39	0.60
	nPSO1 C=20	-0.11	0.27	0.42	0.46	-0.52	0.07	0.45	0.60	-0.83	-0.07	0.39	0.60
	nPSO2 C=5	-0.19	0.30	0.41	0.43	-0.49	0.31	0.54	0.62	-0.73	0.23	0.51	0.65
	nPSO2 C=10	-0.17	0.26	0.40	0.43	-0.50	0.22	0.48	0.59	-0.75	0.10	0.46	0.61
	nPSO2 C=15	-0.15	0.26	0.41	0.44	-0.51	0.16	0.46	0.58	-0.77	0.09	0.43	0.60
	nPSO2 C=20	-0.14	0.25	0.42	0.45	-0.51	0.13	0.46	0.58	-0.79	0.04	0.42	0.59
	nPSO3 C=5	-0.18	0.23	0.40	0.44	-0.48	0.19	0.48	0.61	-0.75	0.09	0.46	0.63
	nPSO3 C=10	-0.14	0.25	0.42	0.46	-0.51	0.11	0.45	0.60	-0.77	0.01	0.42	0.61
	nPSO3 C=15	-0.12	0.26	0.41	0.46	-0.52	0.08	0.46	0.60	-0.80	-0.03	0.40	0.60
	nPSO3 C=20	-0.11	0.26	0.42	0.46	-0.52	0.07	0.46	0.60	-0.82	-0.07	0.39	0.61
m=15	PSO	-0.04	0.25	0.36	0.39	-0.30	0.20	0.52	0.64	-0.48	0.09	0.50	0.68
	nPSO1 C=5	-0.08	0.24	0.35	0.37	-0.35	0.23	0.48	0.59	-0.50	0.20	0.50	0.63
	nPSO1 C=10	-0.05	0.24	0.36	0.38	-0.36	0.18	0.47	0.58	-0.51	0.14	0.47	0.62
	nPSO1 C=15	-0.05	0.25	0.37	0.39	-0.36	0.17	0.48	0.60	-0.53	0.10	0.45	0.62
	nPSO1 C=20	-0.05	0.25	0.36	0.39	-0.35	0.16	0.49	0.60	-0.54	0.07	0.45	0.62
	nPSO2 C=5	-0.10	0.26	0.36	0.36	-0.34	0.31	0.52	0.59	-0.48	0.28	0.54	0.65
	nPSO2 C=10	-0.07	0.26	0.36	0.37	-0.36	0.25	0.48	0.58	-0.50	0.22	0.50	0.62
	nPSO2 C=15	-0.07	0.25	0.35	0.38	-0.36	0.22	0.47	0.58	-0.52	0.17	0.48	0.61
	nPSO2 C=20	-0.07	0.25	0.36	0.38	-0.36	0.20	0.49	0.58	-0.53	0.14	0.47	0.61
	nPSO3 C=5	-0.08	0.24	0.35	0.37	-0.35	0.23	0.48	0.59	-0.50	0.19	0.50	0.63
	nPSO3 C=10	-0.05	0.24	0.36	0.39	-0.35	0.18	0.48	0.59	-0.52	0.13	0.47	0.62
	nPSO3 C=15	-0.05	0.25	0.36	0.39	-0.36	0.17	0.48	0.60	-0.52	0.10	0.46	0.62
	nPSO3 C=20	-0.04	0.25	0.36	0.39	-0.34	0.16	0.49	0.61	-0.53	0.07	0.46	0.63
m=20	PSO	-0.03	0.22	0.29	0.31	-0.18	0.26	0.52	0.63	-0.35	0.17	0.52	0.67
	nPSO1 C=5	-0.05	0.20	0.29	0.30	-0.27	0.26	0.49	0.58	-0.39	0.24	0.50	0.62
	nPSO1 C=10	-0.03	0.22	0.29	0.31	-0.26	0.23	0.49	0.58	-0.41	0.18	0.48	0.61
	nPSO1 C=15	-0.03	0.21	0.29	0.31	-0.25	0.23	0.50	0.59	-0.41	0.15	0.48	0.62
	nPSO1 C=20	-0.03	0.22	0.29	0.31	-0.23	0.24	0.51	0.60	-0.41	0.14	0.48	0.63
	nPSO2 C=5	-0.06	0.22	0.29	0.29	-0.23	0.31	0.51	0.58	-0.38	0.30	0.55	0.63
	nPSO2 C=10	-0.05	0.22	0.29	0.30	-0.27	0.28	0.49	0.57	-0.40	0.26	0.51	0.61
	nPSO2 C=15	-0.04	0.22	0.29	0.30	-0.27	0.26	0.49	0.58	-0.40	0.21	0.49	0.61
	nPSO2 C=20	-0.04	0.22	0.29	0.30	-0.27	0.25	0.50	0.58	-0.41	0.20	0.48	0.61
	nPSO3 C=5	-0.05	0.21	0.29	0.30	-0.27	0.26	0.49	0.58	-0.39	0.23	0.50	0.62
	nPSO3 C=10	-0.03	0.21	0.29	0.31	-0.26	0.23	0.50	0.59	-0.40	0.18	0.48	0.63
	nPSO3 C=15	-0.03	0.21	0.29	0.31	-0.25	0.23	0.50	0.60	-0.40	0.15	0.48	0.62
	nPSO3 C=20	-0.03	0.22	0.29	0.31	-0.23	0.24	0.51	0.60	-0.41	0.14	0.49	0.63

Figure 11. Small-worldness (*ω*) of the PSO and nPSO networks.

Synthetic networks have been generated using the nPSO model with parameters *N* = [100, 500, 1000], *m* = [5, 10, 15, 20], *T* = [0, 0.3, 0.6, 0.9], *γ* = 3 and angular coordinates sampled according to mixture distributions of three different kinds with components *C* = [5, 10, 15, 20]: Gaussian mixture with equal proportions (nPSO1), Gaussian mixture with random proportions (nPSO2), Gaussian and Gamma mixture with equal proportions (nPSO3). For more details on the parameters of the mixture distributions please refer to the Methods. Furthermore, synthetic networks have been generated also using the PSO model with the same parameters *N*, *m*, *T* and *γ*. For each combination of parameters, 10 networks have been generated and the small-worldness (*ω*) has been computed. The heatmap reports for each parameter combination the mean small-worldness (*ω*), coloured according to a blue-to-red colormap in the range [-2, 2].



		N=100				N=500				N=1000			
		T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9	T=0	T=0.3	T=0.6	T=0.9
m=5	PSO	0.00	0.00	0.03	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO1 C=5	0.01	0.00	0.00	0.08	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO1 C=10	0.01	0.00	0.00	0.06	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00
	nPSO1 C=15	0.01	0.00	0.09	0.02	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
	nPSO1 C=20	0.00	0.03	0.08	0.04	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00
	nPSO2 C=5	0.08	0.01	0.05	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.03	0.00
	nPSO2 C=10	0.01	0.03	0.02	0.02	0.00	0.03	0.00	0.00	0.00	0.00	0.02	0.00
	nPSO2 C=15	0.01	0.00	0.02	0.09	0.01	0.00	0.00	0.00	0.03	0.00	0.00	0.00
	nPSO2 C=20	0.00	0.01	0.04	0.01	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.02
	nPSO3 C=5	0.01	0.01	0.02	0.07	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.02
	nPSO3 C=10	0.00	0.02	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO3 C=15	0.00	0.00	0.05	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO3 C=20	0.00	0.02	0.08	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00
m=10	PSO	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO1 C=5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO1 C=10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO1 C=15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO1 C=20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO2 C=5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO2 C=10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO2 C=15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO2 C=20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO3 C=5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO3 C=10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO3 C=15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO3 C=20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
m=15	PSO	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO1 C=5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO1 C=10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO1 C=15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO1 C=20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO2 C=5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO2 C=10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO2 C=15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO2 C=20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO3 C=5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO3 C=10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO3 C=15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO3 C=20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
m=20	PSO	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO1 C=5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO1 C=10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO1 C=15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO1 C=20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO2 C=5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO2 C=10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO2 C=15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO2 C=20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO3 C=5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO3 C=10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO3 C=15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	nPSO3 C=20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 12. Rich-clubness (p-value) of the PSO and nPSO networks.
Synthetic networks have been generated using the nPSO model with parameters $N = [100, 500, 1000]$, $m = [5, 10, 15, 20]$, $T = [0, 0.3, 0.6, 0.9]$, $\gamma = 3$ and angular coordinates sampled according to mixture distributions of three different kinds with components $C = [5, 10, 15, 20]$: Gaussian mixture with equal proportions (nPSO1), Gaussian mixture with random proportions (nPSO2), Gaussian and Gamma mixture with equal proportions (nPSO3). For more details on the parameters of the mixture distributions please refer to the Methods. Furthermore, synthetic networks have been generated also using the PSO model with the same parameters N, m, T and γ. For each combination of parameters, 10 networks have been generated and the rich-clubness (p-value) has been computed. The heatmap reports for each parameter combination the mean rich-clubness (p-value), coloured according to a non-uniform red-to-blue colormap in the range [0, 1] with higher resolution in the range [0, 0.1].

Table 1. Percentage of target-distances greater than the radius of the hyperbolic disk. Synthetic networks have been generated using the PSO and nPSO models with parameters N = [100, 500, 1000], m = [10, 12, 14], T = [0.1, 0.3, 0.5], γ = 3 and, for the nPSO model, angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components C = [4, 8]. For each combination of parameters, 10 networks have been generated using the implementation 3. During the generative procedure of every network, for each step $t \in [m+2, N]$ we computed the percentage of hyperbolic distances (from the new node to the $t-1$ target nodes) greater than the current radius of the hyperbolic disk, and then we averaged over the generative steps. The aim is to understand whether the condition $h_{ij} > R_i$ of Equation (1) is frequently verified. The table reports for each combination of parameters the mean percentage over the 10 networks. Note that the first $m+1$ generative steps have been discarded because the connections are directly established with all the existing nodes, without any computation of the hyperbolic distances.

		PSO			nPSO C = 4			nPSO C = 8		
		T=0.1	*T=0.3*	*T=0.5*	*T=0.1*	*T=0.3*	*T=0.5*	*T=0.1*	*T=0.3*	*T=0.5*
N=100	*m=10*	72.2	75.6	82.3	69.6	73.1	79.4	72.4	75.2	81.2
	m=12	68.5	72.5	79.4	66.7	70.2	77.0	68.7	72.5	78.6
	m=14	65.2	69.9	77.6	63.9	67.5	74.4	65.7	69.9	77.4
N=500	*m=10*	91.6	92.7	94.6	89.2	90.6	92.9	90.1	91.3	93.3
	m=12	90.4	91.6	93.8	87.9	89.3	91.7	88.9	90.2	92.5
	m=14	89.3	90.5	93.0	86.7	88.2	91.0	87.7	89.1	91.6
N=1000	*m=10*	95.1	95.8	96.9	93.6	94.3	95.7	94.1	94.7	95.9
	m=12	94.4	95.1	96.4	92.6	93.4	95.0	93.2	94.0	95.4
	m=14	93.7	94.5	95.9	91.8	92.7	94.5	92.5	93.4	94.9

Table 2. Clustering coefficient comparison for different implementations.

Synthetic networks have been generated using the PSO and nPSO models with parameters $N = [100, 500, 1000]$, $m = [10, 12, 14]$, $T = [0.1, 0.3, 0.5]$, $\gamma = 3$ and, for the nPSO model, angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components $C = [4, 8]$. For each combination of parameters, 10 networks have been generated using the 3 different implementations and the clustering coefficient has been computed. The table reports for each combination of parameters the mean clustering coefficient over the 10 networks generated using the implementation 1. Instead, for the implementations 2 and 3, the difference in the mean clustering coefficient with respect to the implementation 1 is reported.

			Implementation 1			Difference Implementations 1-2			Difference Implementations 1-3		
			T=0.1	*T=0.3*	*T=0.5*	*T=0.1*	*T=0.3*	*T=0.5*	*T=0.1*	*T=0.3*	*T=0.5*
PSO	*N=100*	*m=10*	0.64	0.51	0.43	-0.01	-0.01	0.00	0.01	0.01	0.02
		m=12	0.66	0.53	0.45	-0.01	-0.01	-0.01	0.00	0.01	0.01
		m=14	0.66	0.55	0.48	-0.01	-0.01	-0.01	0.01	0.01	0.01
	N=500	*m=10*	0.65	0.48	0.32	0.00	-0.01	0.00	0.01	0.01	0.01
		m=12	0.65	0.48	0.33	-0.01	-0.01	0.00	0.01	0.01	0.01
		m=14	0.65	0.48	0.34	0.00	-0.01	0.00	0.01	0.01	0.01
	N=1000	*m=10*	0.65	0.48	0.31	0.00	-0.01	0.00	0.01	0.01	0.01
		m=12	0.65	0.48	0.31	0.00	0.00	0.00	0.01	0.02	0.01
		m=14	0.66	0.49	0.32	0.00	0.00	0.00	0.01	0.02	0.01
nPSO C=4	*N=100*	*m=10*	0.62	0.51	0.42	-0.01	-0.01	-0.02	0.01	0.01	0.00
		m=12	0.64	0.53	0.46	-0.01	-0.01	-0.01	0.01	0.01	0.01
		m=14	0.66	0.55	0.49	-0.01	-0.01	-0.02	0.01	0.01	0.01
	N=500	*m=10*	0.50	0.39	0.29	-0.01	-0.01	-0.01	0.01	0.00	0.00
		m=12	0.52	0.41	0.31	0.00	0.00	0.00	0.01	0.01	0.01
		m=14	0.53	0.42	0.32	0.00	0.00	0.00	0.01	0.01	0.00
	N=1000	*m=10*	0.49	0.38	0.26	0.00	0.00	-0.01	0.00	0.01	0.00
		m=12	0.49	0.38	0.27	-0.01	-0.01	0.00	0.01	0.01	0.01
		m=14	0.50	0.39	0.28	0.00	0.00	0.00	0.01	0.01	0.00
nPSO C=8	*N=100*	*m=10*	0.65	0.51	0.43	-0.01	-0.02	0.00	-0.01	0.01	0.01
		m=12	0.66	0.54	0.45	-0.01	0.00	-0.01	0.00	0.02	0.01
		m=14	0.68	0.55	0.47	0.00	-0.02	-0.02	0.00	0.01	0.00
	N=500	*m=10*	0.54	0.43	0.31	0.00	0.00	0.00	0.01	0.01	0.00
		m=12	0.56	0.44	0.33	0.00	0.00	-0.01	0.01	0.01	0.01
		m=14	0.57	0.46	0.34	0.00	-0.01	-0.01	0.01	0.01	0.01
	N=1000	*m=10*	0.51	0.40	0.28	-0.01	-0.01	0.00	0.00	0.01	0.01
		m=12	0.52	0.41	0.29	0.00	0.00	0.00	0.01	0.01	0.01
		m=14	0.53	0.41	0.30	0.00	-0.01	0.00	0.00	0.01	0.01

Table 3. Characteristic path length comparison for different implementations.
Synthetic networks have been generated using the PSO and nPSO models with parameters N = [100, 500, 1000], m = [10, 12, 14], T = [0.1, 0.3, 0.5], γ = 3 and, for the nPSO model, angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components C = [4, 8]. For each combination of parameters, 10 networks have been generated using the 3 different implementations and the characteristic path length has been computed. The table reports for each combination of parameters the mean characteristic path length over the 10 networks generated using the implementation 1. Instead, for the implementations 2 and 3, the difference in the mean characteristic path length with respect to the implementation 1 is reported.

			Implementation 1			Difference Implementations 1-2			Difference Implementations 1-3		
			T=0.1	*T=0.3*	*T=0.5*	*T=0.1*	*T=0.3*	*T=0.5*	*T=0.1*	*T=0.3*	*T=0.5*
PSO	*N=100*	*m=10*	1.9	1.9	1.9	0.0	0.0	0.0	0.0	0.0	0.0
		m=12	1.8	1.8	1.8	0.0	0.0	0.0	0.0	0.0	0.0
		m=14	1.7	1.7	1.7	0.0	0.0	0.0	0.0	0.0	0.0
	N=500	*m=10*	2.5	2.5	2.4	0.0	0.0	0.0	0.0	0.0	0.0
		m=12	2.4	2.4	2.3	0.0	0.0	0.0	0.0	0.0	0.0
		m=14	2.3	2.3	2.2	0.0	0.0	0.0	0.0	0.0	0.0
	N=1000	*m=10*	2.8	2.7	2.7	0.0	0.0	0.0	0.0	0.0	0.0
		m=12	2.6	2.6	2.5	0.0	0.0	0.0	0.0	0.0	0.0
		m=14	2.6	2.5	2.5	0.0	0.0	0.0	0.0	0.0	0.0
nPSO C=4	*N=100*	*m=10*	1.9	1.9	1.9	0.0	0.0	0.0	0.0	0.0	0.0
		m=12	1.8	1.8	1.8	0.0	0.0	0.0	0.0	0.0	0.0
		m=14	1.8	1.8	1.8	0.0	0.0	0.0	0.0	0.0	0.0
	N=500	*m=10*	2.6	2.5	2.5	0.0	0.0	0.0	0.0	0.0	0.0
		m=12	2.5	2.4	2.4	0.0	0.0	0.0	0.0	0.0	0.0
		m=14	2.4	2.4	2.3	0.0	0.0	0.0	0.0	0.0	0.0
	N=1000	*m=10*	2.8	2.8	2.7	0.0	0.0	0.0	0.0	0.0	0.0
		m=12	2.7	2.7	2.6	0.0	0.0	0.0	0.0	0.0	0.0
		m=14	2.6	2.6	2.5	0.0	0.0	0.0	0.0	0.0	0.0
nPSO C=8	*N=100*	*m=10*	1.9	1.9	1.9	0.0	0.0	0.0	0.0	0.0	0.0
		m=12	1.8	1.8	1.8	0.0	0.0	0.0	0.0	0.0	0.0
		m=14	1.7	1.7	1.7	0.0	0.0	0.0	0.0	0.0	0.0
	N=500	*m=10*	2.6	2.6	2.5	0.0	0.0	0.0	0.0	0.0	0.0
		m=12	2.5	2.4	2.4	0.0	0.0	0.0	0.0	0.0	0.0
		m=14	2.4	2.3	2.3	0.0	0.0	0.0	0.0	0.0	0.0
	N=1000	*m=10*	2.9	2.8	2.7	0.0	0.0	0.0	0.0	0.0	0.0
		m=12	2.7	2.7	2.6	0.0	0.0	0.0	0.0	0.0	0.0
		m=14	2.7	2.6	2.5	0.0	0.0	0.0	0.0	0.0	0.0

Table 4. Assortativity comparison for different implementations.

Synthetic networks have been generated using the PSO and nPSO models with parameters N = [100, 500, 1000], m = [10, 12, 14], T = [0.1, 0.3, 0.5], γ = 3 and, for the nPSO model, angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components C = [4, 8]. For each combination of parameters, 10 networks have been generated using the 3 different implementations and the assortativity has been computed. The table reports for each combination of parameters the mean assortativity over the 10 networks generated using the implementation 1. Instead, for the implementations 2 and 3, the difference in the mean assortativity with respect to the implementation 1 is reported.

			Implementation 1			**Difference Implementations 1-2**			**Difference Implementations 1-3**		
			T=0.1	*T=0.3*	*T=0.5*	*T=0.1*	*T=0.3*	*T=0.5*	*T=0.1*	*T=0.3*	*T=0.5*
PSO	*N=100*	*m=10*	-0.16	-0.13	-0.09	0.00	0.00	0.00	0.00	-0.01	-0.01
		m=12	-0.17	-0.13	-0.09	0.01	0.01	0.00	0.00	0.00	-0.01
		m=14	-0.17	-0.13	-0.09	0.01	0.01	0.00	0.00	0.00	0.00
	N=500	*m=10*	-0.06	-0.06	-0.05	0.00	0.00	0.00	0.00	0.00	0.00
		m=12	-0.07	-0.07	-0.05	0.00	0.00	0.00	0.00	0.00	0.00
		m=14	-0.08	-0.08	-0.06	0.00	0.00	0.00	0.00	0.00	0.00
	N=1000	*m=10*	-0.04	-0.04	-0.03	0.00	0.00	0.00	0.00	0.00	0.00
		m=12	-0.05	-0.05	-0.04	0.00	0.00	0.00	0.00	0.00	0.00
		m=14	-0.05	-0.06	-0.04	0.00	0.00	0.00	0.00	0.00	0.00
nPSO C=4	*N=100*	*m=10*	-0.10	-0.09	-0.05	0.01	0.00	0.00	-0.01	0.00	0.00
		m=12	-0.12	-0.09	-0.05	0.01	0.00	0.00	0.00	0.00	0.01
		m=14	-0.13	-0.10	-0.06	0.00	-0.01	0.00	0.00	0.00	-0.01
	N=500	*m=10*	-0.04	-0.04	-0.02	0.00	0.00	0.00	0.01	0.00	0.00
		m=12	-0.05	-0.05	-0.02	0.00	-0.01	-0.01	0.00	0.00	0.00
		m=14	-0.05	-0.05	-0.02	0.00	0.00	0.00	0.00	0.00	0.00
	N=1000	*m=10*	-0.04	-0.03	-0.02	0.00	0.00	0.00	0.00	0.00	0.00
		m=12	-0.04	-0.03	-0.03	0.00	0.00	0.00	0.00	0.00	0.00
		m=14	-0.04	-0.04	-0.02	0.00	0.00	0.00	0.00	0.00	0.00
nPSO C=8	*N=100*	*m=10*	-0.15	-0.12	-0.08	0.00	0.00	0.00	0.01	0.00	0.00
		m=12	-0.17	-0.13	-0.08	0.00	0.00	0.00	0.00	-0.01	0.00
		m=14	-0.18	-0.13	-0.08	0.00	0.00	0.01	0.00	0.00	0.01
	N=500	*m=10*	-0.02	-0.02	-0.02	0.00	0.00	0.00	0.00	0.01	0.00
		m=12	-0.03	-0.03	-0.02	0.00	0.00	0.00	0.00	0.00	0.00
		m=14	-0.04	-0.04	-0.02	0.00	0.00	0.00	0.00	0.00	0.00
	N=1000	*m=10*	-0.02	-0.02	-0.01	0.00	0.00	0.00	0.00	0.00	0.00
		m=12	-0.02	-0.02	-0.01	0.00	0.00	0.00	0.00	0.00	0.00
		m=14	-0.02	-0.02	-0.01	0.00	0.00	0.00	0.00	0.00	0.00

Table 5. LCP-correlation comparison for different implementations.

Synthetic networks have been generated using the PSO and nPSO models with parameters N = [100, 500, 1000], m = [10, 12, 14], T = [0.1, 0.3, 0.5], γ = 3 and, for the nPSO model, angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components C = [4, 8]. For each combination of parameters, 10 networks have been generated using the 3 different implementations and the LCP-correlation has been computed. The table reports for each combination of parameters the mean LCP-correlation over the 10 networks generated using the implementation 1. Instead, for the implementations 2 and 3, the difference in the mean LCP-correlation with respect to the implementation 1 is reported.

			Implementation 1			**Difference Implementations 1-2**			**Difference Implementations 1-3**		
			T=0.1	*T=0.3*	*T=0.5*	*T=0.1*	*T=0.3*	*T=0.5*	*T=0.1*	*T=0.3*	*T=0.5*
PSO	*N=100*	*m=10*	0.99	0.97	0.96	0.00	0.00	0.00	0.00	0.00	0.00
		m=12	0.99	0.98	0.97	0.00	0.00	0.00	0.00	0.00	0.00
		m=14	0.99	0.98	0.97	0.00	0.00	0.00	0.00	0.00	0.00
	N=500	*m=10*	0.98	0.96	0.93	0.00	0.00	0.00	0.00	0.00	0.00
		m=12	0.98	0.97	0.94	0.00	0.00	0.00	0.00	0.00	0.00
		m=14	0.99	0.97	0.94	0.00	0.00	0.00	0.00	0.00	0.00
	N=1000	*m=10*	0.98	0.96	0.92	0.00	0.00	0.00	0.00	0.00	0.01
		m=12	0.98	0.97	0.93	0.00	0.00	0.00	0.00	0.00	0.00
		m=14	0.98	0.97	0.93	0.00	0.00	0.00	0.00	0.00	0.00
nPSO C=4	*N=100*	*m=10*	0.98	0.97	0.96	0.00	0.00	0.00	0.00	0.00	0.00
		m=12	0.99	0.98	0.96	0.00	0.00	0.00	0.00	0.00	0.00
		m=14	0.99	0.98	0.97	0.00	0.00	0.00	0.00	0.00	0.00
	N=500	*m=10*	0.96	0.94	0.92	0.00	0.00	-0.01	0.00	0.00	0.00
		m=12	0.97	0.95	0.93	0.00	0.00	0.00	0.00	0.00	0.00
		m=14	0.97	0.96	0.94	0.00	0.00	0.00	0.00	0.00	0.00
	N=1000	*m=10*	0.95	0.94	0.91	0.00	0.00	-0.01	0.00	0.00	0.00
		m=12	0.96	0.94	0.92	0.00	0.00	0.00	0.00	0.00	0.01
		m=14	0.96	0.95	0.92	0.00	0.00	0.00	0.00	0.00	0.00
nPSO C=8	*N=100*	*m=10*	0.99	0.97	0.96	0.00	0.00	0.00	0.00	0.00	0.00
		m=12	0.99	0.98	0.96	0.00	0.00	0.00	0.00	0.00	0.00
		m=14	0.99	0.98	0.97	0.00	0.00	0.00	0.00	0.00	0.00
	N=500	*m=10*	0.97	0.95	0.93	0.00	0.00	0.00	0.00	0.00	0.00
		m=12	0.97	0.96	0.93	0.00	0.00	0.00	0.00	0.00	0.00
		m=14	0.98	0.96	0.94	0.00	0.00	0.00	0.00	0.00	0.00
	N=1000	*m=10*	0.96	0.95	0.91	0.00	0.00	0.00	0.00	0.00	0.00
		m=12	0.97	0.95	0.92	0.00	0.00	0.00	0.00	0.00	0.00
		m=14	0.97	0.96	0.93	0.00	0.00	0.00	0.00	0.00	0.01

Table 6. Power-law exponent (γ) comparison for different implementations.
Synthetic networks have been generated using the PSO and nPSO models with parameters *N* = [100, 500, 1000], *m* = [10, 12, 14], *T* = [0.1, 0.3, 0.5], *γ* = 3 and, for the nPSO model, angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components *C* = [4, 8]. For each combination of parameters, 10 networks have been generated using the 3 different implementations and the power-law exponent has been computed. The table reports for each combination of parameters the mean power-law exponent over the 10 networks generated using the implementation 1. Instead, for the implementations 2 and 3, the difference in the mean power-law exponent with respect to the implementation 1 is reported.

			Implementation 1			**Difference Implementations 1-2**			**Difference Implementations 1-3**		
			T=0.1	*T=0.3*	*T=0.5*	*T=0.1*	*T=0.3*	*T=0.5*	*T=0.1*	*T=0.3*	*T=0.5*
PSO	*N=100*	*m=10*	2.9	2.9	2.9	0.1	-0.1	0.0	0.0	0.0	-0.1
		m=12	2.9	2.9	2.9	0.0	0.0	-0.1	0.0	0.0	0.0
		m=14	2.9	2.9	2.9	0.0	0.0	0.0	0.0	0.0	0.0
	N=500	*m=10*	2.9	2.9	2.9	0.0	0.0	0.0	0.0	0.0	0.0
		m=12	2.9	2.9	2.9	0.0	0.0	0.0	0.0	0.0	0.0
		m=14	3.0	2.9	2.9	0.0	0.0	0.0	0.1	0.0	0.0
	N=1000	*m=10*	2.9	2.9	2.9	-0.1	0.0	0.0	0.0	0.0	0.0
		m=12	2.9	2.9	2.9	0.0	0.0	0.0	0.0	0.0	0.0
		m=14	2.9	2.9	2.9	0.0	0.0	0.0	0.0	0.0	0.0
nPSO C=4	*N=100*	*m=10*	3.3	3.0	2.9	0.3	-0.1	0.1	0.2	0.1	0.0
		m=12	3.2	3.1	2.9	0.1	0.1	-0.1	0.0	0.1	-0.1
		m=14	3.2	3.0	2.9	0.0	-0.1	0.0	0.0	0.0	-0.2
	N=500	*m=10*	2.9	2.8	2.8	0.0	0.0	0.0	0.0	0.0	0.0
		m=12	2.8	2.8	2.8	0.0	0.0	0.0	0.0	0.0	0.0
		m=14	2.9	2.9	2.9	0.0	0.0	0.0	0.0	0.1	0.0
	N=1000	*m=10*	2.9	2.9	2.8	0.0	0.0	0.0	0.0	0.0	0.0
		m=12	2.9	2.9	2.8	0.0	0.0	0.0	0.0	0.0	0.0
		m=14	2.9	2.9	2.8	0.0	0.0	0.0	0.0	0.0	0.0
nPSO C=8	*N=100*	*m=10*	3.0	2.9	2.9	0.0	0.0	0.0	0.1	0.1	0.0
		m=12	3.0	2.9	3.0	0.0	0.0	0.1	0.0	0.0	0.2
		m=14	3.0	2.9	3.0	0.0	0.0	0.0	0.0	0.0	0.1
	N=500	*m=10*	2.8	2.8	2.8	0.0	-0.2	-0.1	0.0	0.0	-0.1
		m=12	2.9	3.0	2.9	-0.1	0.0	0.0	-0.1	0.1	0.0
		m=14	2.9	2.9	3.0	-0.1	-0.1	0.0	0.0	-0.2	0.0
	N=1000	*m=10*	2.8	2.8	2.9	0.0	0.0	0.0	0.0	0.0	0.0
		m=12	2.9	2.8	2.8	0.0	0.0	0.0	0.0	0.0	0.0
		m=14	2.8	2.9	2.8	0.0	0.0	0.0	0.0	0.0	0.0

Paper C: Leveraging the nonuniform PSO network model as a benchmark for performance evaluation in community detection and link prediction

Alessandro Muscoloni[1] and Carlo Vittorio Cannistraci[1,2,*]

[1] Biomedical Cybernetics Group, Biotechnology Center (BIOTEC), Center for Molecular and Cellular Bioengineering (CMCB), Center for Systems Biology Dresden (CSBD), Department of Physics, Technische Universität Dresden, Tatzberg 47/49, 01307 Dresden, Germany; [2] Brain bio-inspired computing (BBC) lab, IRCCS Centro Neurolesi "Bonino Pulejo", Messina, Italy; * Correspondence should be addressed to: kalokagathos.agon@gmail.com

Published: *New Journal of Physics*, 20, 063022.

Abstract

Advances in network geometry pointed out that structural properties observed in networks derived from real complex systems can emerge in the hyperbolic space (HS). The nonuniform PSO (nPSO) is a generative model recently introduced in order to grow random geometric graphs in the HS, reproducing networks that have realistic features such as high clustering, small-worldness, scale-freeness and rich-clubness, with the additional possibility to control the community organization.

Generative models allowing to tune the structural properties of 'realistic' synthetic networks are fundamental, because they offer a ground-truth to investigate how predictive algorithms react to controlled topological variations. Here, we discuss how to leverage the nPSO model as a synthetic benchmark to compare the performance of methods for community detection and link prediction; and we prove that the nPSO offers a reliable and realistic testing framework which can complement other existing benchmarks not based on latent geometry.

Furthermore, we confirm that network embedding information can improve community detection, whereas boosting link prediction in HS still needs further investigations. Indeed, we find that the presence of communities in nPSO significantly modifies the performance of link predictors and is fundamental for the reproducibility of results observed on real networks. The nPSO can trigger valuable insights to understand the intrinsic rules of link-growth and self-organization that connect topology to geometry and that are encoded in link-prediction algorithms differentiating their performance.

C.1 Introduction

Several generative models have been proposed in the past years as synthetic benchmarks for community detection [40]. One of the first benchmark has been developed by Girvan and Newman [6], where all the nodes have the same degree and all the communities identical size, and has been later extended by Danon et al. [41] in order to generate communities with different size. A generative model able to reproduce structural properties closer to the ones observed in real networks has been proposed by Lancichinetti-Fortunato-Radicchi (LFR) [42], which is characterized by a power-law distribution of degree and community size. This model turned out to be a particular version of the degree-corrected stochastic block model [40], [119]. The LFR benchmark has been later extended to directed and weighted networks with overlapping communities [120], meaning that communities can have also nodes in common [40]. A further benchmark for overlapping communities has been introduced by Ball et al. [43], however, having all the nodes the same expected degree, it is less realistic and flexible than the LFR [40]. Interestingly, all these well-known models of networks with communities are not generated according to a latent geometry, which instead will be the fundamental theme of this study.

The investigation of hidden geometrical spaces behind complex network topologies has been a fervid topic in recent years and, currently, the hyperbolic space seems to be one of the most appropriate in order to explain many of the structural features observed in real networks [15], [17]–[20], [30], [32], [51], [65], [68], [74], [81]. The PSO [20] is a generative model that grows random geometric graphs in the hyperbolic space, reproducing networks that have realistic features such as clustering, small-worldness, scale-freeness and rich-clubness. However, real networks exhibit another very important feature that is community organization, not contemplated in the original PSO model. For such reason, Muscoloni et al. [121] introduced a variation of it, the nonuniform PSO (nPSO) model, which allows to explicitly control the number of communities, their size and mixing property. Providing a benchmark for community detection requires the possibility to manipulate structural properties such as average node degree, clustering, small-worldness and scale-freeness, in order to assess how differently community detection algorithms react to these controlled topological variations. The nPSO can have a large impact not only in this scenario, but also in other real applications such as link prediction. For instance, by generating ground-truth synthetic networks with the nPSO, it is possible to investigate the extent to which the community organization affects, together with other topological properties, the performance of link prediction algorithms. This, in turn, can advocate the comprehension of the intrinsic rules of network wiring that connect topology to geometry and that are encrypted in algorithms for link prediction differentiating their performance.

In this study we will discuss how the nPSO model can be employed to: a) test the performance of state-of-the-art community detection methods and verify the agreement with previous studies; b) test whether the community detection performance can be improved using information gained by the network embedding in the hyperbolic space; c) test the performance of state-of-the-art link prediction methods on PSO and nPSO networks and verify the importance of community organization for link prediction and the agreement with respect to real networks; d) discuss the extent to which link prediction can be improved using information gained by the network embedding in the hyperbolic space.

C.2 Results and Discussion

Comparison of community detection algorithms on nPSO networks

The community detection algorithms Louvain [44], Infomap [45], Walktrap [46] and Label propagation [47] are four state-of-the-art approaches that have been shown to provide high performances on synthetic benchmarks [48]–[50]. Louvain [44] is a model-free and unsupervised heuristic method based on modularity optimization; Infomap [45] finds the community structure by minimizing the expected description length of a random walker trajectory using the Huffman coding process; Walktrap [46] is based on an agglomerative method for hierarchical clustering, where the similarities between the nodes or groups are obtained using random walks; Label propagation [47] is an iterative algorithm in which each node label is updated with the one owned by the majority of the neighbours until reaching a consensus (see Methods for details). Louvain and Infomap have been also recently tested on small-size and large-size real networks, resulting overall among the best performing on recovering ground-truth communities associated to metadata [114]. In this study, we compare these four community detection approaches across synthetic networks generated using diverse nPSO parameter combinations: N = [100, 500, 1000] (network size), m = [10, 12, 14] (half of average node degree), T = [0.1, 0.3, 0.5] (temperature, inversely related to clustering), γ = 3 (power-law degree distribution exponent) and angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components C = [4, 8]. The values chosen for the parameter m are centred around the average m (which is equal to 12 rounded) computed on the dataset of small-size real networks. The values chosen for N and T are intended to cover the range of network size and clustering coefficient observed in the dataset of small-size real networks. Since the average γ estimated on the dataset of small-size real networks is higher than the typical range $2 < \gamma < 3$, we choose $\gamma = 3$.

Figure 1 reports the mean Normalized Mutual Information (NMI) [122] performance (NMI is a measure to assess the performance of community detection, see Methods for details) and related standard errors (10 network repetitions considered for each parameter combination) of the community detection algorithms applied to nPSO networks with 4 communities. The results indicate that overall Louvain appears as the strongest approach, with an almost perfect detection over different values of network size, average node degree and temperature. Infomap highlights problems in correctly detecting the communities when there are too many inter-community links, as can be seen for $N = 100$ and increasing temperature. The higher temperature in fact leads to a higher number of links between nodes that are geometrically far in the disk, which increases the mixing between the communities. The performance is more stable for bigger networks, although in general slightly worse than Louvain. Walktrap results as robust as Louvain to the increase of network temperature, but the NMI is slightly lower for $N = 100$ and $N = 1000$. As last, Label propagation, which is the fastest approach, but the one with lowest accuracy, performs worse than the other methods and presents the same problem as Infomap for $N = 100$. This issue has been already pointed out in the study of Yang et al. [48], in which it is shown that for a high mixing of the communities Louvain and Walktrap are more robust, whereas Infomap and Label propagation tend to drop in performance. Hence, the nPSO model here proposed seems to provide a good benchmark to test community detection algorithms on networks generated using a latent geometry model which is based on the hyperbolic space.

Figure 2 reports the NMI performance on networks with 8 communities. Focusing firstly on the performance on bigger networks (N = [500-1000]), it can be noticed that Louvain and Infomap swap their behaviour, with Infomap going close to the perfect community detection and Louvain slightly decreasing its performance. Walktrap, instead, remains quite robust and slightly improves for $N = 1000$. Label propagation still remains the most unstable, although surpassing Louvain for very low temperature. In the study of Yang et al. [48] it is shown that, when the mixing of the communities is not high, Louvain can slightly underestimate the number of communities for networks of increasing size, which might be the reason of its reduced performance in large networks (N = [500-1000]) with respect to the other approaches. Focusing now on the small size networks ($N = 100$), Suppl. Figure 1 highlights that the methods preserve the same ranking with respect to the case with 4 communities, but they all decrease their performance. The reason is that, being the network small and keeping the average node degree constant, the increase of the communities leads to more inter-community links, making the community structure less detectable.

Community detection on nPSO networks using network embedding information

Recently, Muscoloni et al. [51] introduced coalescent embedding, a model-free topological-based machine learning class of algorithms that exploits nonlinear unsupervised dimensionality reduction to infer the node coordinates in the hyperbolic space. The study also demonstrates that, exploiting the geometrical embedding information in order to weight the adjacency matrix in input to the community detection algorithms, the performance of the respective unweighted variants can be improved [51]. Since the evaluation tests in Muscoloni et al. [51] were executed on real network datasets, it remains the doubt that the performance evaluation was not objective and possibly biased by the restricted metadata available on the community annotation. This is a perfect example for clarifying the utility of the nPSO model. We would like to have a dataset of realistic networks for which we know the ground-truth community organization and that we can use as a benchmark to test whether the community detection algorithms benefit from being applied according to the geometrical embedding information in the hyperbolic space. Therefore, we repeat the same tests made by Muscoloni et al. [51] on real networks, but here we use nPSO synthetic networks as benchmark. More precisely, each network is embedded in the hyperbolic space using the coalescent embedding techniques and the inverse hyperbolic distances (HD) between the nodes are used to weight the observed links of the input network for the Louvain, Infomap, Walktrap and Label propagation algorithms. In a second variant (working only for the Louvain method) the non-observed links are also weighted, using the inverse hyperbolic shortest paths (HSP) (see Methods for details). The results of these tests are in Tables 1-2 and Suppl. Tables 1-2, which report the mean NMI for nPSO networks with 4 and 8 communities, the mean NMI over all the networks and the mean ranking, comparing the weighted variants and the unweighted ones.

Table 1 displays for the Louvain algorithm the mean NMI and the mean ranking over all the networks, which are indicators of the general performance on many parameter combinations of the nPSO. Almost all the HD-based methods increase their performance with respect to the unweighted Louvain, going from a mean NMI of 0.89 up to 0.93. A few HSP-based methods are also able to reach similar performances, whereas most of the others exhibit a decrease in NMI. Figure 1 provides a more detailed comparison between the unweighted Louvain and the top-ranked of the weighted variants (RA-LE-HD) for nPSO networks with 8 communities, showing the NMI performance in all the network configurations. The improvement is consistent over all the parameter combinations and becomes more pronounced for bigger networks, remarkably bringing the NMI close to a perfect detection.

Table 2 displays the mean NMI and the mean ranking results for the Infomap algorithm and shows that all the HD-based methods increase the mean NMI with respect to the unweighted Infomap, going from a value of 0.74 up to 0.87, but not all of them obtain a higher ranking. The reason is explained in Figure 2, which provides a more detailed comparison between the unweighted Infomap and the top-ranked of the weighted variants (EBC-ncISO-HD) for nPSO networks with 8 communities. As already discussed in the previous section, Infomap highlights problems in correctly detecting the communities when there are too many inter-community links, as can be seen for $N = 100$ and increasing temperature. In this scenario, the increase in NMI given by the HD-weighting is remarkable. For bigger networks, instead, the detection using the unweighted Infomap is almost perfect and the HD-based variants tend to obtain the same performance, except for a slight decrease at $N = 1000$ and $T = 0.1$.

Suppl. Table 1 displays the results for the Walktrap algorithm and shows that all the HD-based methods obtain a higher ranking, although the mean NMI has only a small increase with respect to the unweighted Walktrap, going from a value of 0.89 up to 0.91.

At last, Suppl. Table 2 displays the results for the Label propagation algorithm and shows that only some of the HD-based methods increase the mean NMI with respect to the unweighted Label propagation. However, the increase can be outstanding, going from a NMI of 0.56 up to 0.70. Indeed, Label propagation is the least accurate among the methods and therefore the one with the largest margin for improvement.

Overall, this investigation on nPSO networks confirms that the performance of community detection algorithms can be improved using network embedding information, in agreement with what has been previously demonstrated on real networks by Muscoloni et al. [51].

Comparison of link prediction algorithms on PSO and nPSO networks

We investigated if the nPSO networks could represent a realistic framework also for testing link prediction algorithms. We compared the performance of state-of-the-art approaches [52], [53] (CRA [54]–[56], RA [57], SPM [58], SBM [59]) across both PSO and nPSO networks generated using diverse parameter combinations. Cannistraci-Resource-Allocation (CRA) is a mechanistic model which implements a local-topology-based parameter-free deterministic rule for topological link-prediction motivated by the local-community-paradigm [54]–[56]; the standard Resource-Allocation (RA) is instead motivated by the resource allocation process; Structural Perturbation Method (SPM) is a global and model-free approach that relies on a theory similar to the first-order perturbation in quantum mechanics [58]; Stochastic Block Model (SBM) is a global approach based on general idea of a block model, where the nodes are partitioned into groups and the probability that two nodes are connected depends only on the groups to which they belong [59] (see Methods for details). The aim of this section is to understand whether using synthetic networks (generated by PSO or nPSO) is possible to replicate the same and diverse link-predictors' performance obtained on real complex networks. In case this is not possible, we would like to check whether the community organization present in the nPSO plays an important role to replicate the results on real networks and to facilitate the performance of some algorithms in comparison to the case of networks where communities are not present, such as the ones generated by classical PSO.

In the evaluation framework adopted, 10% of the links are randomly removed from the network and used to test the precision of the link predictors in recovering them. Every method is executed on the network deprived of the randomly removed links and provides a ranking of candidate links. The precision is computed as the proportion of links correctly recovered considering a number of top-ranked candidate links equivalent to the 10% removed. The percentage of removed links is kept at 10% in order to ensure that the original topological structure of the network is minimally perturbed and the topological properties are conserved. This removal and re-prediction procedure is repeated multiple times (100 repetitions for all the methods, only 10 repetitions for SBM due to the high computational time) and the average precision is considered to evaluate the performance of each link predictor.

In the PSO networks, as reported in Figure 5, the three methods CRA, SPM and RA obtain a comparable precision, with RA performing slightly better (in particular for $N = 100$) than CRA, which in turn offers a small improvement with respect to SPM for low temperature $T = 0.1$. SBM is slightly inferior to the other predictors for $N = 100$, and it decreases performance for larger networks. The fact that RA performs slightly better than CRA in wiring-prediction of synthetic networks generated by a uniform model without communities is expected, because RA does not account for local-community-organization in the network, and therefore it should adhere better than CRA to the community-free structure of the PSO. In the PSO model a node connects to the other nodes in an 'isotropic' manner, it means that there is not any connection preference on the left or right side of the angular coordinates. Whereas, in the nPSO model a node connects to the other nodes in an 'anisotropic' manner, it means that there is a preference for nodes on the left or right side of the angular coordinates in relation to the direction of localization of the community to which that node belongs. In practice, RA is a weighted version of common neighbours similarity that penalizes each common neighbour for its degree ‘isotropically’ (in the sense that: taking a common neighbour the penalization is the same for each link that contributes to determine its degree). Instead, CRA penalizes each common neighbour for its degree ‘anisotropically’ (in the sense that: taking a common neighbour the penalization is not effective for links that - although contribute to determine its degree - are connected to other common neighbours and create a local community). Hence, if our rationale is correct, we expect that CRA should clearly outperform RA not only on nPSO but also on real networks, and this improvement should emerge especially for growing network size, because a large number of nodes favours the generation of non-uniform topology.

When the link-prediction techniques are tested on the nPSO networks, as shown in Figures 6-7, it can be noticed that the introduction of the communities leads to a different scenario. For N = 100 (small size networks) the performance of CRA, SPM and RA remains overall comparable, with SBM slightly inferior. For increasing network size, and particularly for low temperature, all the methods reduce their precision. However, the decrease is smaller for SBM and CRA, leading to a different ranking, where CRA overcomes the other link prediction approaches regardless of the number of communities adopted (4 or 8). In order to check whether these results resemble a real scenario, we tested the link prediction algorithms also on real networks (SBM only on small-size networks due to the high computational time complexity). Suppl. Tables 3-5 summarize the performance of the approaches for small-size and large-size real networks, both in predicting randomly removed links and in predicting links with time information. For small-size networks SPM obtains the highest mean precision-ranking, followed by CRA, SBM and, as last, RA. This result does not approximate well what has been seen in the artificial networks, and the main reason can be that the networks present different characteristics. In particular, most of the small-size real networks do not have a marked power-lawness, as reported in Suppl. Table 6, whereas the PSO and nPSO networks, at least in their basic implementation, are designed to follow a power-law degree distribution. We emphasize that in the other study in which we theoretically introduce and discuss the nPSO model [121], it has been numerically proven that changing or removing the community structure in the nPSO networks (while keeping fixed the other model parameters) does not significantly affect the main structural features of the network, like clustering coefficient, characteristic path length, power-law exponent, assortativity and LCP-correlation [121]. Looking at the large-size networks, which tend to be scale-free (see Suppl. Table 7), CRA obtains the best mean precision-ranking, reproducing the results reported for increasing network size on the nPSO networks, which, therefore, seems to offer a more realistic framework with respect to the original PSO model. SBM performance is in general comparable to the one of the other algorithms only on small-size networks ($N = 100$), whereas for increasing network size SBM considerably loses performance showing a clear problem of inference.

Link prediction using network embedding information

While the previous section focused on link prediction approaches exploiting only topological information, we are now going to discuss whether adopting also information gained by the embedding in the hyperbolic space could be beneficial.

The plots in Figures 5-7, previously commented, report as a reference also the performance that is obtained if the links are predicted ranking them by the hyperbolic distances (HD) or the hyperbolic shortest paths (HSP), computed using the original coordinates of the nodes and without considering the links randomly removed. It can be noticed that in the PSO model the HD performance is slightly lower than CRA for $T = 0.1$ and higher than CRA for $T = [0.3, 0.5]$, whereas in the nPSO model, with the introduction of the communities, the HD performance consistently decreases, since most of the non-existing links within each community are at low HD and therefore are top-ranked. The HSP, instead, provide a quite low precision and always lower than the other methods, which could be expected because the links in the PSO and nPSO networks are established only depending on the HD.

Since the real networks tend to present a community structure, these results obtained on the nPSO suggest that embedding a network in the hyperbolic space and using the ranked HD for predicting the links will not generally lead to high values of precision.

In order to prove this we applied the coalescent embedding techniques [123], a topological-based machine learning class of algorithms that provides a fast and efficient hyperbolic embedding, and the other main hyperbolic embedding methods: HyperMap-CN [25] and LPCS [124]. We employ here the same procedure of link prediction evaluation adopted above. The small-size real networks are embedded in the hyperbolic space after the 10% random link removal. Given the geometrical coordinates of the embedded nodes, we adopted both the HD-ranking and HSP-ranking to re-predict the randomly removed links, the results are reported in Suppl. Table 8. The maximum average precision offered by the techniques is 0.17, which confirms the expectations independently from the mapping method used. However, if the HSP-ranking is adopted, we notice a general increase of performance with respect to the HD-ranking. From an applied standpoint, this result suggests that on real networks combining both the geometrical and the topological information (using the HSP) might help to improve the prediction, and actually this is a confirmation of a result already presented by Cannistraci et al. [36] on prediction of protein interactions by network embedding. At last, we underlines that the small difference in performance between HD and HSP detected in the real networks is better resembled by the nPSO model. In fact, using the PSO model, the difference in performance between HD and HSP is large, and this is markedly in disagreement with the results obtained in real networks.

As discussed for the nPSO networks, we argue that most of the non-existing links within the communities would be characterized by a low HD or HSP, and therefore they tend to be top-ranked, reducing the likelihood to predict the randomly removed (missing) links, which are not necessarily inside the community. In order to provide an explanatory example for this point, we focused on three small-size real networks for which community metadata are available, and we analysed the top-ranked links provided by the hyperbolic embedding techniques after the random link removal. Suppl. Table 9 reports for each method and each network the percentage of intra-community candidate-links predicted in the top-ranking and the percentage of these which corresponds to correct predictions. The results support our argument showing that on average ~80% of the top-ranked candidate-links are intra-community, and that among these only ~10% represent correct predictions.

To conclude, we would like to discuss the fact that in the generative procedure of the PSO and nPSO models the links are deterministically established between the closest nodes only for the case $T = 0$, whereas for $T > 0$ with a probability dependent on the hyperbolic distance. Since real networks are hardly characterized by a temperature zero, we argue that the usage of the HD-ranking might be not the best solution for link prediction. Ideally, the connection probability as a function of the hyperbolic distance should be empirically estimated from the length gained by the links when the network is embedded in the hyperbolic disk. Assuming that this well approximates the original connection probability function of the unknown generative model, the hyperbolic distances between non-connected nodes can be converted in connection probabilities according to the probability distribution function inferred from the existing links. Alternatively, as second option, the hyperbolic distances between the nodes could be converted in connection probabilities according to the mathematical formula of the model, after having fitted the temperature parameter of the network. Therefore, instead to adopt a greedy procedure that ranks (smallest values come first) the candidate links according to their distances in the hyperbolic space, a correct procedure should sample the candidate links respecting the empirically estimated connection probability of the networked system. However, since the main focus of the article is not link prediction, we procrastinate this investigation to future studies.

C.3 Conclusion

In this work, the nPSO model [121] has been used as a benchmark for testing four state-of-the-art community detection approaches, and the evaluations through several parameter combinations highlighted two main points: firstly, the communities are always detected with high accuracy by at least one method; secondly, performance limitations arisen in particular conditions for some community detection methods are in agreement with findings produced in previous studies. These points represent the first main result of our study and suggest that the model can be adopted as a valid framework for community detection studies. It is not necessarily intended to replace the previous benchmarks, but potentially to complement them, in particular the LFR model [42] that seems the most realistic among the ones not based on latent geometry.

Furthermore, making tests on nPSO networks, we confirmed that community detection algorithms can be boosted if their input is not a simple unweighted adjacency matrix but a pre-weighted one, where the strength of each link is the inverse of the respective distance in the hyperbolic space. While this has been previously demonstrated on real networks [51], a confirmation on networks with ground-truth community structure (such as nPSO) was necessary because the metadata on the community structure of real networks remains always questionable. Therefore, this represents the second main result of this study.

At last, we tested state-of-the-art link prediction algorithms both in real and artificial networks, and we showed that the ranking of the methods according to their performance on the nPSO model is closer to the one in the real networks with respect to the PSO model.

In addition, we highlighted that embedding a network in the hyperbolic space and adopting the HD-ranking for suggesting the links more likely to appear will not lead generally to an efficient prediction, pointing out that the usage of a weighted sampling of candidate links according to empirically estimated connection probabilities needs to be investigated.

To conclude, we propose the nPSO model as a valid framework able to generate realistic networks that can be adopted, among the many possibilities, as a reliable benchmark for testing algorithms designed for community detection and link prediction. Although several benchmarks for community detection have been proposed in previous studies, at the best of our knowledge the nPSO model is the first one able to produce realistic networks with a tailored community structure based on a latent geometrical space, and the first able to reproduce results for link prediction evaluation reasonably close to the ones obtained on real networks.

C.4 Methods

1. Community detection

1.1 Louvain

The Louvain algorithm [44] is separated into two phases, which are repeated iteratively.

At first every node in the (weighted) network represents a community in itself. In the first phase, for each node i, it considers its neighbours j and evaluates the gain in modularity that would take place by removing *i* from its community and placing it in the community of j. The node *i* is then placed in the community j for which this gain is maximum, but only if the gain is positive. If no gain is possible node *i* stays in its original community. This process is applied until no further improvement can be achieved.

In the second phase the algorithm builds a new network whose nodes are the communities found in the first phase, whereas the weights of the links between the new nodes are given by the sum of the weight of the links between nodes in the corresponding two communities. Links between nodes of the same community lead to self-loops for this community in the new network.

Once the new network has been built, the two phase process is iterated until there are no more changes and a maximum of modularity has been obtained. The number of iterations determines the height of the hierarchy of communities detected by the algorithm.

For each hierarchical level there is a possible partition to compare to the ground truth annotation. In this case, the hierarchical level considered is the one that guarantees the best match, therefore the detected partition that gives the highest NMI value.

We used the R function *multilevel.community*, an implementation of the method available in the *igraph* package [96].

In this study, the embedding of the network in the hyperbolic space has been exploited in order to weight the input adjacency matrix. Given the hyperbolic coordinates, the observed links have been weighted using the formula

$$x_{ij}^{HD} = \frac{1}{1 + HD_{ij}}$$

where HD_{ij} is the hyperbolic distance between nodes i and j. For the Louvain algorithm a further variant has been tested in which also the non-observed links have been weighted using the formula

$$x_{ij}^{HSP} = \frac{1}{1 + HSP_{ij}}$$

where HSP_{ij} is the hyperbolic shortest path between nodes i and j, computed as the sum of the hyperbolic distances over the shortest path.

1.2. Infomap

The Infomap algorithm [45] finds the community structure by minimizing the expected description length of a random walker trajectory using the Huffman coding process [97].

It uses the hierarchical map equation (a further development of the map equation, to detect community structures on more than one level) in the form $L(M) = q_{\curvearrowleft}H(Q) + \sum_{i=1}^{m} L(M^{i})$. $L(M)$ is the lower bound of the code length to specify a network path of a partitioning M, $q_{\curvearrowleft}H(Q)$ is the Shannon information at the coarsest level of the partitioning, $L(M^{i}) = q_{\circlearrowleft}^{i}H(Q^{i}) + \sum_{j=1}^{m^{i}} L(M^{ij})$ is the lower bound of the code length to specify a network path of a partitioning M at sublevel i and $L(M^{ij\ldots k}) = p_{\circlearrowleft}^{ij\ldots k}H(P^{ij\ldots k})$ is the lower bound of the code length at the finest modular level with submap $M^{ij\ldots k}$.

The hierarchical map equation indicates the theoretical limit of how concisely a network path can be specified using a given partition structure. In order to calculate the optimal partition (community) structure, this limit can be computed for different partitions and the community annotation that gives the shortest path length is chosen.

For each hierarchical level there is a possible partition to compare to the ground truth annotation. In this case, the hierarchical level considered is the one that guarantees the best match, therefore the detected partition that gives the highest NMI value.

We used the C implementation released by the authors at http://www.mapequation.org/code.html.

In this study, the embedding of the network in the hyperbolic space has been exploited in order to weight the input adjacency matrix. Given the hyperbolic coordinates, the observed links have been weighted using the hyperbolic distances (HD) as described for Louvain.

1.3. Walktrap

The Walktrap algorithm [46] is based on an agglomerative method for hierarchical clustering: the nodes are iteratively grouped into communities exploiting the similarities between them. The nodes similarities are obtained using random walks and are based on the idea that random walks tend to get trapped into densely connected subgraphs corresponding to communities.

The agglomerative method uses heuristics to choose which communities to merge and implements an efficient way to update the distances between communities. At the end of the procedure a hierarchy of communities is obtained and each level offers a possible partition. The algorithm chooses as final result the partition that maximizes the modularity.

We used the R function *walktrap.community*, an implementation of the method available in the *igraph* package [96].

1.4. Label propagation

The label propagation algorithm [47] initializes each node with a unique label and iteratively updates each node label with the one owned by the majority of the neighbours, with ties broken uniformly at random. The update is performed in an asynchronous way and the order of the nodes at each iteration is chosen randomly. As the labels propagate through the network, densely connected groups of nodes quickly reach a consensus on a unique label. The iterative process stops when every node has the same label as the majority its neighbours, ties included. At the end of the procedure the nodes having the same label are grouped together to form a community. Since the aim is not the optimization of an objective function and the propagation process contains randomness, there are more possible partitions that satisfy the stop criterion and therefore the solution is not unique.

We used the R function *label.propagation.community*, an implementation of the method available in the *igraph* package [96].

1.5. Normalized Mutual Information

The evaluation of the community detection has been performed using the Normalized Mutual Information (NMI) as in [98]. The entropy can be defined as the information contained in a distribution *p(x)* in the following way:

$$H(X) = \sum_{x \in X} p(x) \log p(x)$$

The mutual information is the shared information between two distributions:

$$I(X,Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p_1(x)p_2(y)}\right)$$

To normalize the value between 0 and 1 the following formula can be applied:

$$NMI = \frac{I(X,Y)}{\sqrt{H(X)H(Y)}}$$

If we consider a partition of the nodes in communities as a distribution (probability of one node falling into one community), we can compute the matching between the annotation obtained by the community detection algorithm and the ground truth communities of a network as follows:

$$H(C_D) = \sum_{h=1}^{n_D} \frac{n_h^D}{N} \log(\frac{n_h^D}{N})$$

$$H(C_T) = \sum_{l=1}^{n_T} \frac{n_l^T}{N} \log(\frac{n_l^T}{N})$$

$$I(C_D, C_T) = \sum_h \sum_l \frac{n_{h,l}}{N} \log\left(\frac{n_{h,l}}{n_h^D n_l^T}\right).$$

$$NMI(C_D, C_T) = \frac{I(C_D, C_T)}{\sqrt{H(C_D)H(C_T)}}$$

Where:

N – number of nodes;

n^D, n^T – number of communities detected by the algorithm (D) or ground truth (T);

$n_{h,l}$ – number of nodes assigned to the h-th community by the algorithm and to the l-th community according to the ground truth annotation.

We used the MATLAB implementation available at http://commdetect.weebly.com/. As suggested in the code, when $\frac{N}{n^T} \leq 100$, the NMI should be adjusted in order to correct for chance [99].

2. Link prediction methods

2.1. Cannistraci-Resource-Allocation (CRA)

Cannistraci-Resource-Allocation (CRA) is a local-based, parameter-free and model-based deterministic rule for topological link-prediction in both monopartite [54] and bipartite networks [55], [56]. It is based on the *local-community-paradigm* (LCP) which is a bioinspired theory recently proposed in order to model local-topology-dependent link-growth in a class of real complex networks characterized by the development of diverse, overlapping and hierarchically organized local-communities [54]. Being a local-community-based method, it assigns to every candidate interaction a likelihood score looking only at the neighbours nodes and their cross-interactions. In particular, the paradigmatic shift introduced by the LCP is to consider not only the common-neighbours of the interacting nodes but also the links between those common-neighbours, which in practice form all together a local-community.

For each candidate interaction between nodes i and j, the score is assigned according to the following equation [54]:

$$CRA(i,j) = \sum_{k \in \Phi(i) \cap \Phi(j)} \frac{|\varphi(k)|}{|\Phi(k)|}$$

Where:

k: common neighbour of nodes i and j

$\Phi(i)$: set of neighbours of node i

$|\Phi(k)|$: cardinality of set $\Phi(k)$, equivalent to the degree of k

$\varphi(k)$: sub-set of neighbours of k that are also common-neighbours of i and j

$|\varphi(k)|$: equivalent to the local-community degree of k (see Fig. 2)

The higher the CRA score, the higher the likelihood that the interaction exists, therefore the candidate interactions are ranked by decreasing CRA scores and the obtained ranking is the link-prediction result. The method has been implemented in MATLAB. The code is available at: https://sites.google.com/site/carlovittoriocannistraci/

2.2. Resource-Allocation (RA)

Resource-Allocation (RA) is a local-based, parameter-free and model-based deterministic rule for topological link-prediction [57], motivated by the resource allocation process taking place in networks. Considering a pair of nodes that are not directly connected, one node can send some resource to the other one through their common-neighbours, which play the role of transmitters. It assumes the simplest case where every transmitter equally distributes a unit of resource between its neighbours. For each candidate interaction between nodes i and j, the score is assigned according to the following equation [57]:

$$RA(i,j) = \sum_{k \in \Phi(i) \cap \Phi(j)} \frac{1}{|\Phi(k)|}$$

Where:

k: common neighbour of nodes i and j

$\Phi(i)$: set of neighbours of node i

$|\Phi(k)|$: cardinality of set $\Phi(k)$, equivalent to the degree of k

The higher the RA score, the higher the likelihood that the interaction exists, therefore the candidate interactions are ranked by decreasing RA scores and the obtained ranking is the link-prediction result. The method has been implemented in MATLAB.

2.3. Structural Perturbation Method (SPM)

SPM is a structural perturbation method that relies on a theory similar to the first-order perturbation in quantum mechanics [58]. Unlike CH, it is a global approach, meaning that it exploits the information of the complete adjacency matrix in order to compute the likelihood score to assign to every candidate interaction. A high-level description of the procedure is the following:

1) Randomly remove a subset of the edges ΔE (usually 10%) from the network adjacency matrix x, obtaining a reduced adjacency matrix x^R.
2) Compute the eigenvalues and eigenvectors of x^R.
3) Considering ΔE as a perturbation of x^R, construct the perturbed matrix $\tilde{x}$ via a first-order approximation that allows the eigenvalues to change while keeping fixed the eigenvectors.
4) Repeat steps 1-3 for 10 independent iterations and take the average of the perturbed matrices $\tilde{x}$.

The idea behind the method is that a missing part of the network is predictable if it does not significantly change the structural features of the observable part, represented by the eigenvectors of the matrix. If this is the case, the perturbed matrices should be good approximations of the original network [58]. The entries of the average perturbed matrix represents the scores for the candidate links. The higher the score the greater the likelihood that the interaction exists, therefore the candidate interactions are ranked by decreasing scores and the obtained ranking represents the link-prediction result.

The MATLAB implementation of the method has been provided by the authors.

2.4 Stochastic Block Model (SBM)

The framework based on stochastic block model (SBM) considered in this study has been introduced by Guimerà et al. [59] in order to identify both missing and spurious interactions in noisy network observations. The general idea of a block model is that the nodes are partitioned into groups and the probability that two nodes are connected depends only on the groups to which they belong. Assuming that there is no prior knowledge about which partition is more suitable for the observed network, the mathematical formula for obtaining the reliability of an individual link between nodes *i* and j is [59]:

$$R_{ij} = \frac{1}{Z} \sum_{p \in P} \left(\frac{l_{\sigma_i \sigma_j} + 1}{r_{\sigma_i \sigma_j} + 2} \right) exp[-H(p)]$$

Where the sum is over every partition p in the space P of all the possible partitions of the network into groups, σ_i is the group of node *i* in partition p, $l_{\alpha\beta}$ is the number of links between groups α and β, $r_{\alpha\beta}$ is the maximum number of possible links between groups α and β. The function *H(p)* is:

$$H(p) = \sum_{\alpha \leq \beta} \left[ln\left(r_{\alpha\beta} + 1\right) + ln\binom{r_{\alpha\beta}}{l_{\alpha\beta}} \right]$$

And the normalization factor is:

$$Z = \sum_{p \in P} exp[-H(p)]$$

However, since the exploration of all the possible partitions is too computationally expensive, the Metropolis algorithm, which is based on a stochastic procedure, is exploited in order to sample only a subset of partitions that are relevant for the estimation of the link reliability [59]. The higher the reliability the greater the likelihood that a non-observed interaction actually exists, therefore the candidate interactions are ranked by decreasing scores and the obtained ranking represents the link-prediction result. The C code of the method has been released by the authors and can be download from the website http://seeslab.info/downloads/network-c-libraries-rgraph/.

2.5. Coalescent embedding

The expression coalescent embedding refers to a topological-based machine learning class of algorithms that exploits nonlinear unsupervised dimensionality reduction to infer the nodes angular coordinates in the hyperbolic space [51]. The techniques are able to perform a fast and accurate mapping of a network in the 2D hyperbolic disk, the 3D hyperbolic sphere, and potentially also in higher dimensions.

The first step of the algorithm for a 2D embedding consists in weighting the network in order to suggest geometrical distances between connected nodes, since it has been shown that improves the mapping accuracy [51]. If the network is unweighted, the topological-based pre-weighting rules repulsion-attraction (RA) or edge-betweenness-centrality (EBC) can be applied. The rules are devised to suggest geometrical distances between the connected nodes, using either local (RA) or global (EBC) topological information [51].

Given the weighted network, the second step consists in performing the nonlinear dimensionality reduction. Two different kinds of machine learning approaches can be used, manifold-based (LE, ISO, ncISO) or Minimum-Curvilinearity-based (MCE, ncMCE). The details about which dimensions of the embedding should be considered are provided in the original publication [51].

In order to assign the angular coordinates in the 2D embedding space, either a circular adjustment or an equidistant angular adjustment (EA) can be considered. The circular adjustment for the manifold-based approaches consists in exploiting directly the polar coordinates of the 2D reduced space, whereas for the Minimum-Curvilinearity-based in rearranging the node points on the circumference following the same ordering of the 1D reduced space and proportionally preserving the distances. Using the equidistant angular adjustment, instead, the nodes are equidistantly arranged on the circumference, which might help to correct for short-range angular noise present in the embedding [51].

The radial coordinates are assigned according to a mathematical formula which takes into account both the position of the nodes in their ranking by degree and the scale-freeness of the node degree distribution [23]. The exponent γ of the power-law degree distribution has been fitted using the MATLAB script *plfit.m*, a procedure described by Clauset et al. [91] and released at http://www.santafe.edu/~aaronc/powerlaws/.

In the link prediction application, the hyperbolic distances (HD) are computed between the nodes in the hyperbolic space and the candidate interactions are ranked by increasing HD, the obtained ranking is the link prediction result.

In a second variant that combines both geometrical and topological information, the network is weighted using the HD and the hyperbolic shortest paths (HSP) are computed as sum of the HD over the shortest path between each pair of nodes. The candidate interactions are ranked by increasing HSP and the obtained ranking is the link prediction result. The method has been implemented in MATLAB.

The code is available at https://github.com/biomedical-cybernetics/coalescent_embedding.

2.6. HyperMap-CN

HyperMap [23] is a method to map a network into the hyperbolic space based on Maximum Likelihood Estimation (MLE). For sake of clarity, the first algorithm for MLE-based network embedding in the hyperbolic space is not HyperMap, but to the best of our knowledge is the algorithm proposed by Boguñá et al. in [18]. HyperMap is basically an extension of that method applied to the PSO model [20]. It replays the hyperbolic growth of the network and at each time step *i* it finds the coordinates of the added node *i* by maximizing the likelihood that the network was produced by the E-PSO model [23]. According to the MLE procedure, the nodes are added in decreasing order of degree. The radial coordinates depend on the time step *i* and on the exponent γ of the power-law degree distribution. The angular coordinates, instead, are assigned by maximizing a likelihood function $L_{i,L}$, with the aim of mapping connected nodes at a low hyperbolic distance and disconnected nodes at a high hyperbolic distance. The maximization is done by numerically trying different angular coordinates in steps of $2\pi/N$ and choosing the one that leads to the biggest $L_{i,L}$.

HyperMap-CN [25] is a further development of HyperMap, where the inference of the angular coordinates is not performed anymore maximizing the likelihood $L_{i,L}$,based on the connections and disconnections of the nodes, but using another local likelihood $L_{i,CN}$, based on the number of common neighbours between each node *i* and the previous nodes j < *i* at final time. Here the hybrid model has been used, a variant of the method in which the likelihood $L_{i,CN}$ is only adopted for the high degree nodes and $L_{i,L}$ for the others, yielding a shorter running time. Furthermore, a speed-up heuristic and corrections steps can be applied. The speed-up can be achieved by getting an initial estimate of the angular coordinate of a node i only considering the previous nodes $j < i$ that are i's neighbours, the maximum likelihood estimation is then performed only looking at an interval around this initial estimate. Correction steps can be used at predefined times i: each existing node $j < i$ is visited and with the knowledge of the rest of the coordinates the angle of j is updated to the value that maximizes the likelihood $L_{j,L}$. The C++ implementation of the method has been released by the authors at the website https://bitbucket.org/dk-lab/2015_code_hypermap. In our simulations, neither correction steps nor speed-up heuristic have been used. The input parameter γ has been fitted as described for the coalescent embedding method. The temperature has been set to a default value *T* = 0.1.

2.7. LPCS

Link Prediction with Community Structure (LPCS) [87] is a hyperbolic embedding technique that consists of the following steps: (1) Detect the hierarchical organization of communities. (2) Order the top-level communities starting from the one that has the largest number of nodes and using the Community Intimacy index, which takes into account the proportion of edges within and between communities. (3) Recursively order the lower level communities based on the order of the higher-level communities, until reaching the bottom level in the hierarchy. (4) Assign to every bottom-level community an angular range of size proportional to the nodes in the community, in order to cover the complete circle with non-overlapping angular ranges. Sample the angular coordinates of the nodes uniformly at random within the angular range of the related bottom-level community. (5) Assign the radial coordinates as described for the MCA method.

The LPCS code firstly takes advantage of the R function *multilevel.community* for detecting the hierarchy of communities, an implementation of the Louvain method [44] available in the *igraph* package [96], while the following embedding steps have been implemented in MATLAB.

3. Networks datasets

3.1. PSO model

The Popularity-Similarity-Optimization (PSO) model [20] is a generative network model recently introduced in order to describe how random geometric graphs grow in the hyperbolic space. In this model the networks evolve optimizing a trade-off between node popularity, abstracted by the radial coordinate, and similarity, represented by the angular coordinate distance, and they exhibit many common structural and dynamical characteristics of real networks.

The model has five input parameters:

- $N > 0$, number of nodes in the network;
- $m > 0$, equal to half of the average node degree;
- $T \geq 0$, network temperature, which controls the network clustering; the network clustering is maximized at $T = 0$, it decreases almost linearly for $T = [0,1)$ and it becomes asymptotically zero if $T > 1$;
- $\beta \in (0, 1]$, popularity fading parameter, or alternatively $\gamma \geq 2$, exponent of the power-law degree distribution, due to the relationship $\gamma = 1 + 1/\beta$;
- $\zeta = \sqrt{-K} > 0$, where *K* is the curvature of the hyperbolic plane. Since changing ζ rescales the node radial coordinates and this does not affect the topological properties of network [20], in the rest of the article we will consider $K = -1$.

Building a network in the hyperbolic disk requires the following steps:

(1) Initially the network is empty;

(2) At time $i = 1, 2, \dots, N$ a new node *i* appears with radial coordinate $r_i = 2ln(i)$ and angular coordinate θ_i uniformly sampled in $[0,2\pi]$; all the existing nodes $j < i$ increase their radial coordinates according to $r_j(i) = \beta r_j + (1 - \beta) r_i$ in order to simulate popularity fading;

(3) If $T = 0$, the new node connects to the *m* hyperbolically closest nodes; if $T > 0$, the new node picks a randomly chosen existing node $j < i$ and, given that it is not already connected to it, it connects to it with probability

$$p(i,j) = \frac{1}{1 + \exp\left(\frac{h_{ij} - R_i}{2T}\right)}$$

repeating the procedure until it becomes connected to *m* nodes.

Note that

$$R_i = r_i - 2\ln\left[\frac{2T\left(1 - e^{-(1-\beta)\ln(i)}\right)}{\sin(T\pi)\, m(1-\beta)}\right]$$

is the current radius of the hyperbolic disk, and

$$h_{ij} = arccosh\left(\cosh r_i \cosh r_j - \sinh r_i \sinh r_j \cos\theta_{ij}\right)$$

is the hyperbolic distance between node *i* and node *j,* where

$$\theta_{ij} = \pi - \left|\pi - \left|\theta_i - \theta_j\right|\right|$$

is the angle between these nodes.

(4) The growing process stops when *N* nodes have been introduced.

3.2. Nonuniform PSO (nPSO) model

The nonuniform PSO model [121] is a variation of the PSO model introduced in order to confer to the generated networks an adequate community structure, which is lacking in the original model. Since the connection probabilities are inversely proportional to the hyperbolic distances, a uniform distribution of the nodes over the hyperbolic disk does not create agglomerates of nodes that are concentrated on angular sectors and that are more densely connected between each other than with the rest of the network. A nonuniform distribution, instead, allows to do it by generating heterogeneity in the angular node arrangement. Given the parameters of the PSO model (N, m, T, γ) and a nonuniform probability distribution defined in [0, 2π[, the procedure to generate a network is the same described in the section for the uniform case, with the only difference that the angular coordinates of the nodes are not sampled uniformly but according to the given nonuniform probability distribution.

In this study, without loss of generality, we will concentrate on Gaussian mixture distributions, with communities that emerge in correspondence of the different components.

A Gaussian mixture distribution is characterized by the following parameters [113]:

- $C > 0$, which is the number of components, each one representative of a community;
- $\mu_{1\ldots C} \in [0{,}2\pi[$, which are the means of the components, representing the central locations of the communities in the angular space;

- $\sigma_{1\ldots C} > 0$, which are the standard deviations of the components, determining how much the communities are spread in the angular space; a low value leads to isolated communities, a high value makes the adjacent communities to overlap;
- $\rho_{1\ldots C}$ $(\sum_i \rho_i = 1)$, which are the mixing proportions of the components, determining the relative sizes of the communities.

Note that, although the means of the components are located in [0, 2π[, the sampling of the angular coordinate θ can fall out of this range. In this case, it has to be shifted within the original range using the modulo operator: $\theta = modulo(\theta, 2\pi)$.

Although the parameters of the Gaussian mixture distribution allow for the investigation of disparate scenarios, as a first case of study we focused on the most straightforward setting. For a given number of components *C*, we considered their means equidistantly arranged over the angular space, the same standard deviation and equal mixing proportions:

- $\mu_i = \frac{2\pi}{C} * (i-1) \quad i = 1 \ldots C$
- $\sigma_1 = \sigma_2 = \ldots = \sigma_C = \sigma$
- $\rho_1 = \rho_2 = \ldots = \rho_C = \frac{1}{C}$

In particular, in our simulations we fixed the standard deviation to 1/6 of the distance between two adjacent means $\left(\sigma = \frac{1}{6} * \frac{2\pi}{C}\right)$, which allowed for a reasonable isolation of the communities independently from their number. The community memberships are assigned considering for each node the component whose mean is at the lowest angular distance.

3.3. Real networks

The real networks have been transformed into undirected and unweighted, self-loops have been removed and the largest connected component has been considered.

Mouse neural: in-vivo single neuron connectome that reports mouse primary visual cortex (layers 1, 2/3 and upper 4) synaptic connections between neurons [125].

Karate: social network of a university karate club collected by Wayne Zachary in 1977. Each node represents a member of the club and each edge represents a tie between two members of the club [88]. Community metadata are available, the communities are formed by a split of the club into two parts, each following one trainer.

Dolphins: a social network of bottlenose dolphins. The nodes are the bottlenose dolphins (genus Tursiops) of a bottlenose dolphin community living off Doubtful Sound, a fjord in New

Zealand. An edge indicates a frequent association. The dolphins were observed between 1994 and 2001 [126].

Macaque neural: a macaque cortical connectome, assembled in previous studies in order to merge partial information obtained from disparate literature and database sources [127].

Polbooks: nodes represent books about US politics sold by the online bookseller Amazon.com. Edges represent frequent co-purchasing of books by the same buyers, as indicated by the "customers who bought this book also bought these other books" feature on Amazon. The network was compiled by V. Krebs and is unpublished, but can found at http://www-personal.umich.edu/~mejn/netdata/. Community metadata are available, the three communities are given by the political orientation of the books as either conservative, neutral or liberal.

ACM2009_contacts: network of face-to-face contacts (active for at least 20 seconds) of the attendees of the ACM Conference on Hypertext and Hypermedia 2009 [128].

Football: network of American football games between Division IA colleges during regular season Fall 2000 [6].

Physicians innovation: the network captures innovation spread among physicians in the towns in Illinois, Peoria, Bloomington, Quincy and Galesburg. The data was collected in 1966. A node represents a physician and an edge between two physicians shows that the left physician told that the right physician is his friend or that he turns to the right physician if he needs advice or is interested in a discussion [129].

Manufacturing email: email communication network between employees of a mid-sized manufacturing company [130].

Littlerock foodweb: food web of Little Rock Lake, Wisconsin in the United States of America. Nodes are autotrophs, herbivores, carnivores and decomposers; links represent food sources [131].

Jazz: collaboration network between Jazz musicians. Each node is a Jazz musician and an edge denotes that two musicians have played together in a band. The data was collected in 2003 [132].

Residence hall friends: friendship network between residents living at a residence hall located on the Australian National University campus [133].

Haggle contacts: contacts between people measured by carried wireless devices. A node represents a person and an edge between two persons shows that there was a contact between them [134].

Worm nervous: a C. *Elegans* connectome representing synaptic interactions between neurons [37].

Netsci: a co-authorship network of scientists working on networks science [135].

Infectious contacts: network of face-to-face contacts (active for at least 20 seconds) of people during the exhibition INFECTIOUS: STAY AWAY in 2009 at the Science Gallery in Dublin [128].

Flightmap: a network of flights between American and Canadian cities [136].

Email: email communication network at the University Rovira i Virgili in Tarragona in the south of Catalonia in Spain. Nodes are users and each edge represents that at least one email was sent [137].

Polblog: a network of front-page hyperlinks between blogs in the context of the 2004 US election. A node represents a blog and an edge represents a hyperlink between two blogs [100]. Community metadata are available, the two communities represent the political opinions of the blogs (conservative or liberal).

Odlis: Online Dictionary of Library and Information Science (ODLIS): ODLIS is designed to be a hypertext reference resource for library and information science professionals, university students and faculty, and users of all types of libraries. Version December 2000 [138].

Advogato: a trust network of the online community platform Advogato for developers of free software launched in 1999. Nodes are users of Advogato and the edges represent trust relationships [139].

Arxiv astroph: collaboration graph of authors of scientific papers from the arXiv's Astrophysics (astro-ph) section. An edge between two authors represents a common publication [140].

Thesaurus: this is the Edinburgh Associative Thesaurus. Nodes are English words, and a directed link from A to B denotes that the word B was given as a response to the stimulus word A in user experiments [141].

Arxiv hepth: this is the network of publications in the arXiv's High Energy Physics – Theory (hep-th) section. The links that connect the publications are citations [140].

Facebook: a network of a small subset of posts to user's walls on Facebook. The nodes of the network are Facebook users, and each edge represents one post, linking the users writing a post to the users whose wall the post is written on [142].

ARK200909-ARK201012: Autonomous systems (AS) Internet topologies extracted from the data collected by the Archipelago active measurement infrastructure (ARK) developed by CAIDA, from September 2009 up to December 2010. The connections in the topology are not physical but logical, representing AS relationships [101].

Most of the networks in the dataset can be downloaded from the Koblenz Network Collection at http://konect.uni-koblenz.de.

Hardware and software

MATLAB code has been used for all the simulations, carried out partly on a workstation under Windows 8.1 Pro with 512 GB of RAM and 2 Intel(R) Xenon(R) CPU E5-2687W v3 processors with 3.10 GHz, and partly in the ZIH-Cluster Taurus of the TU Dresden.

Funding

Work in the CVC laboratory was supported by the independent research group leader starting grant of the Technische Universität Dresden. AM was partially supported by the funding provided by the Free State of Saxony in accordance with the Saxon Scholarship Program Regulation, awarded by the Studentenwerk Dresden based on the recommendation of the board of the Graduate Academy of TU Dresden.

Acknowledgements

We thank Alexander Mestiashvili and the BIOTEC System Administrators for their IT support, Claudia Matthes for the administrative assistance and the Centre for Information Services and High Performance Computing (ZIH) of the TUD.

Author contributions

CVC conceived the study and designed the numerical experiments, AM implemented the code and performed the computational analysis. Both the authors analysed and interpreted the results. AM wrote the draft of the article according to CVC suggestions and CVC corrected and improved it to arrive to the final draft. CVC designed the figures and AM realized them. CVC planned, directed and supervised the study.

Competing interests

The authors declare no competing financial interests.

C.5 Figures and Tables

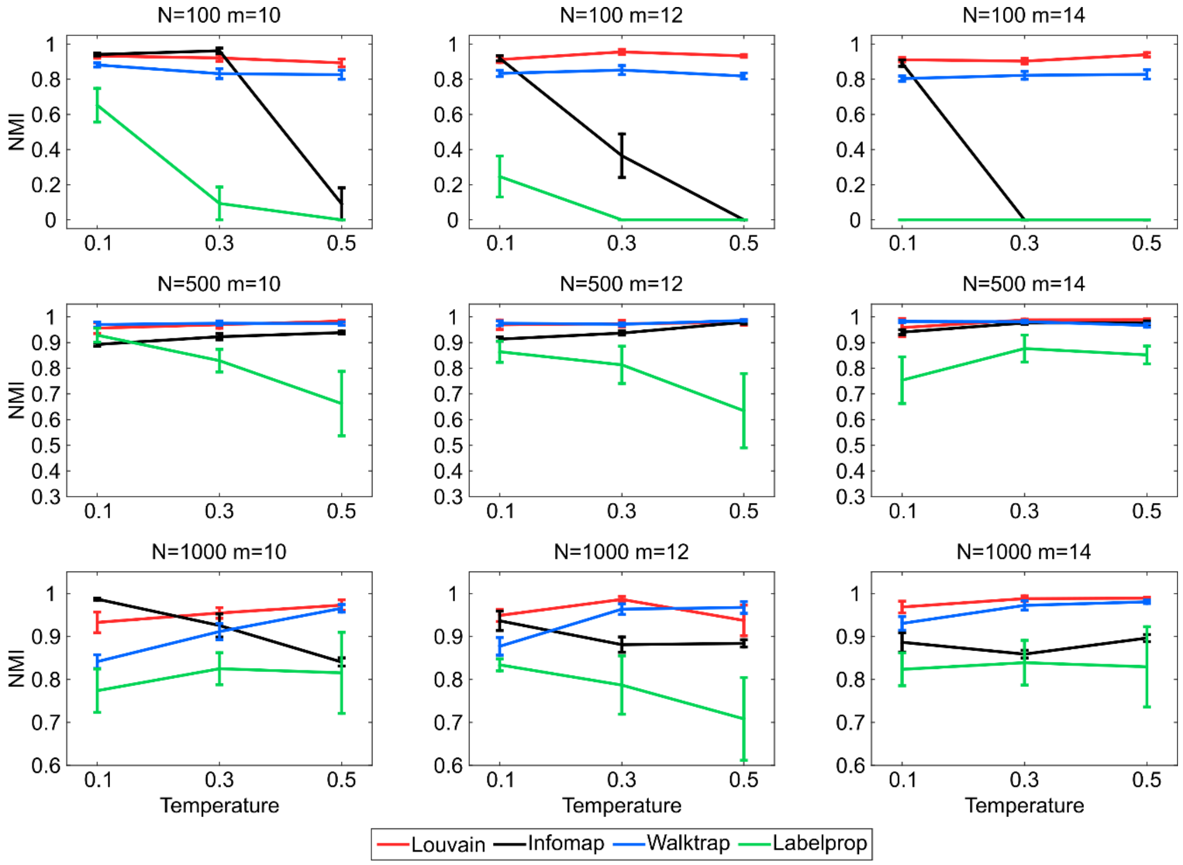


Figure 1. Community detection on nPSO networks with 4 communities.
Synthetic networks have been generated using the nPSO model with parameters N = [100, 500, 1000], m = [10, 12, 14], T = [0.1, 0.3, 0.5], γ = 3 and angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components C = 4. For each combination of parameters, 10 networks have been generated. For each network the community detection methods Louvain, Infomap, Walktrap and Label propagation have been executed and the communities detected have been compared to the annotated ones computing the Normalized Mutual Information (NMI). The plots report for each parameter combination the mean NMI and standard error over the random repetitions.

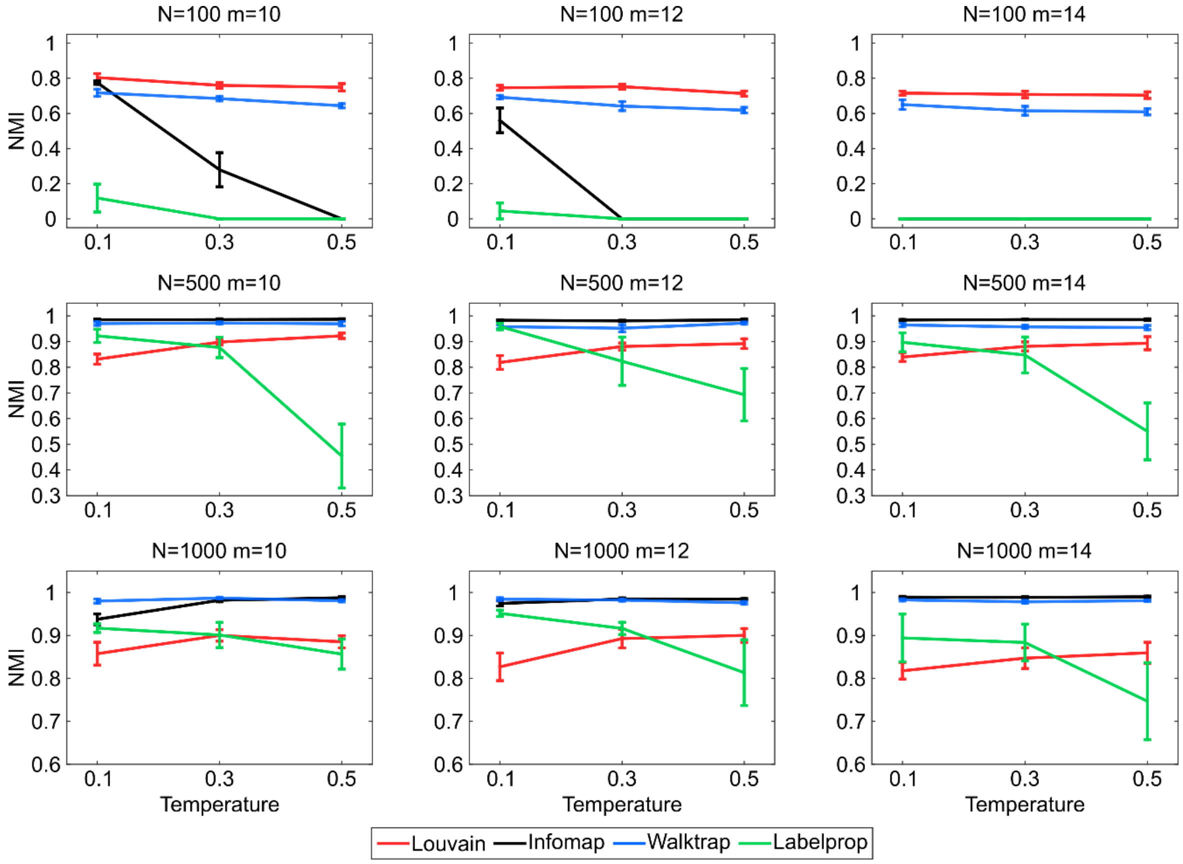


Figure 2. Community detection on nPSO networks with 8 communities.
Synthetic networks have been generated using the nPSO model with parameters $N = [100, 500, 1000]$, $m = [10, 12, 14]$, $T = [0.1, 0.3, 0.5]$, $\gamma = 3$ and angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components $C = 8$. For each combination of parameters, 10 networks have been generated. For each network the community detection methods Louvain, Infomap, Walktrap and Label propagation have been executed and the communities detected have been compared to the annotated ones computing the Normalized Mutual Information (NMI). The plots report for each parameter combination the mean NMI and standard error over the random repetitions.

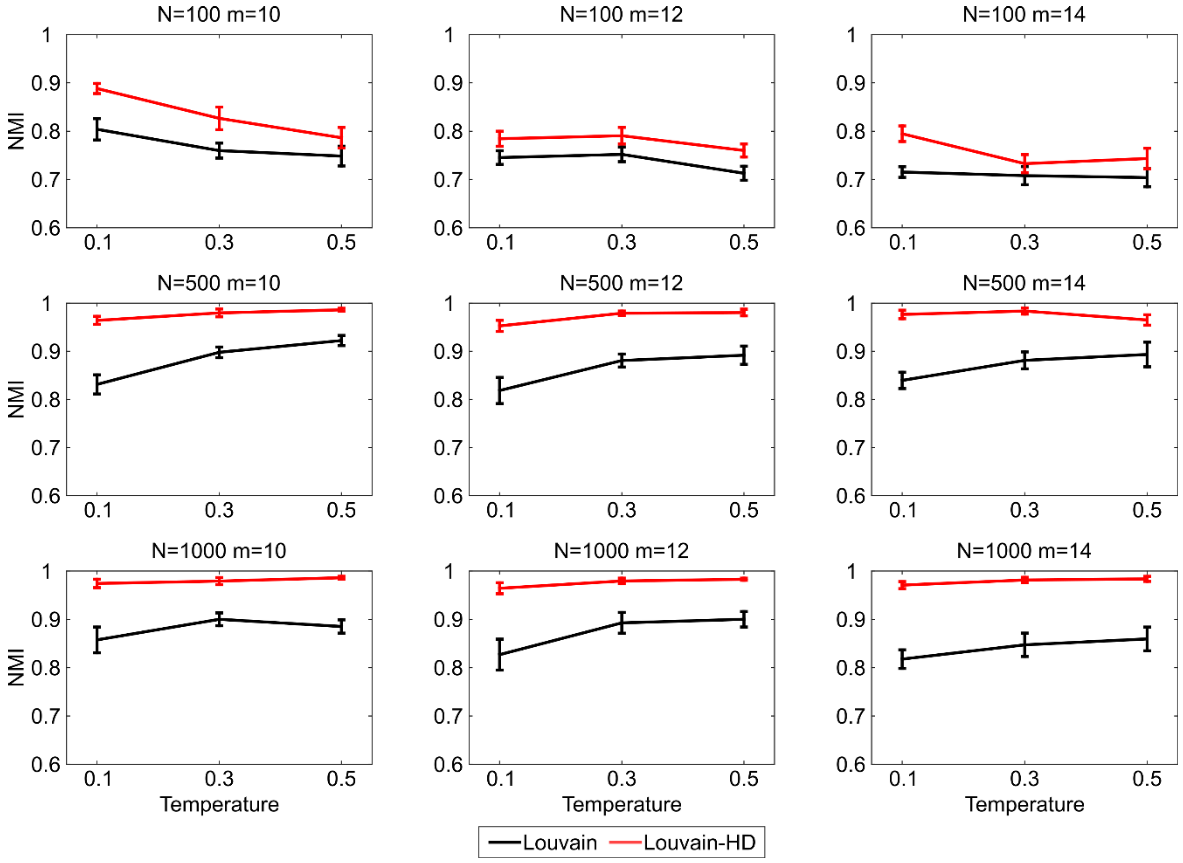


Figure 3. Community detection on nPSO networks (C = 8) using Louvain-HD.
Synthetic networks have been generated using the nPSO model with parameters N = [100, 500, 1000], m = [10, 12, 14], T = [0.1, 0.3, 0.5], γ = 3 and angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components C = 8. For each combination of parameters, 10 networks have been generated. Each network has been embedded in the hyperbolic space using the best performing coalescent embedding technique (RA-LE-HD) and the embedding coordinates are used to weight the input matrix for the Louvain algorithm: observed links are weighted using the hyperbolic distances (HD) between the nodes (see Methods for details). As a reference, the Louvain algorithm has been run giving in input also the unweighted adjacency matrix. The communities detected have been compared to the annotated ones computing the Normalized Mutual Information (NMI). The plots report for each parameter combination the mean NMI and standard error over the random repetitions. The comparison for all the coalescent embedding methods is reported in Table 1.

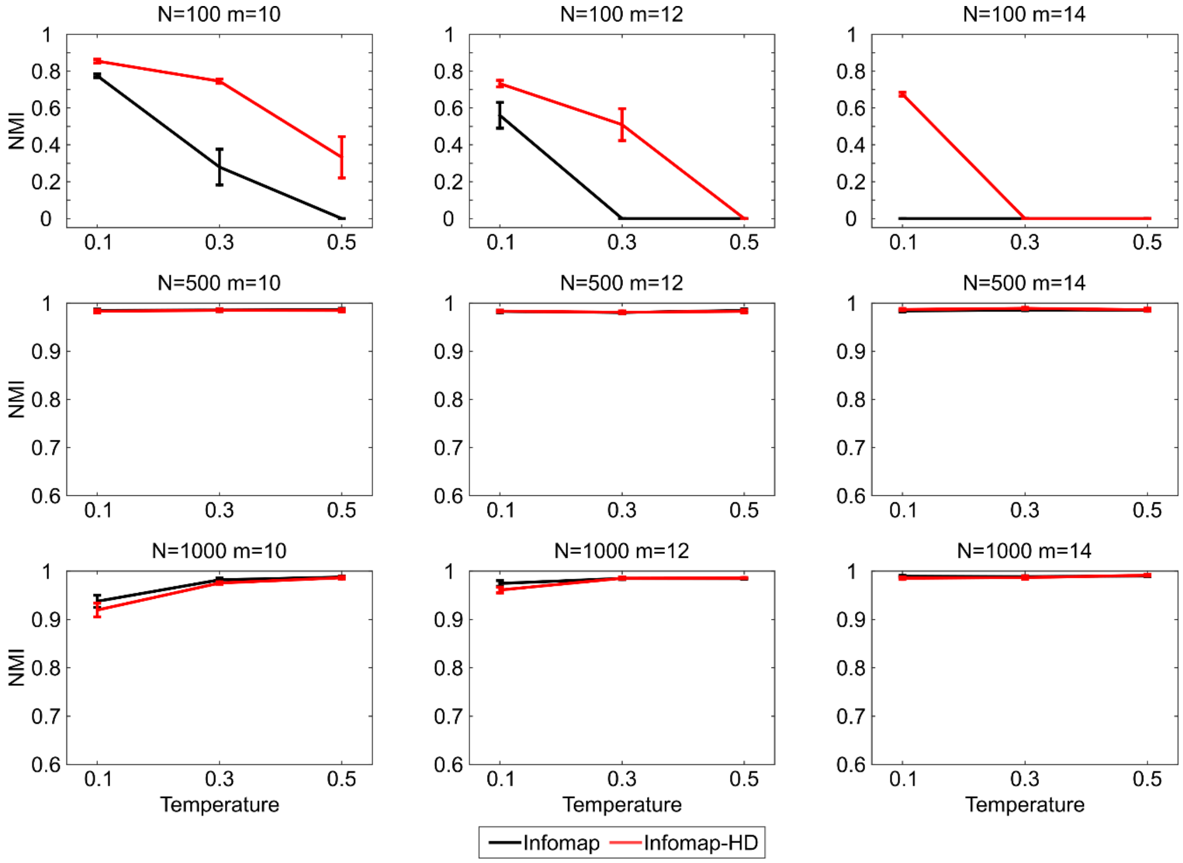


Figure 4. Community detection on nPSO networks (*C* = 8) using Infomap-HD.
Synthetic networks have been generated using the nPSO model with parameters $N = [100, 500, 1000]$, $m = [10, 12, 14]$, $T = [0.1, 0.3, 0.5]$, $\gamma = 3$ and angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components $C = 8$. For each combination of parameters, 10 networks have been generated. Each network has been embedded in the hyperbolic space using the best performing coalescent embedding technique (EBC-ncISO-HD) and the embedding coordinates are used to weight the input matrix for the Infomap algorithm: observed links are weighted using the hyperbolic distances (HD) between the nodes (see Methods for details). As a reference, the Infomap algorithm has been run giving in input also the unweighted adjacency matrix. The communities detected have been compared to the annotated ones computing the Normalized Mutual Information (NMI). The plots report for each parameter combination the mean NMI and standard error over the random repetitions. The comparison for all the coalescent embedding methods is reported in Table 2.

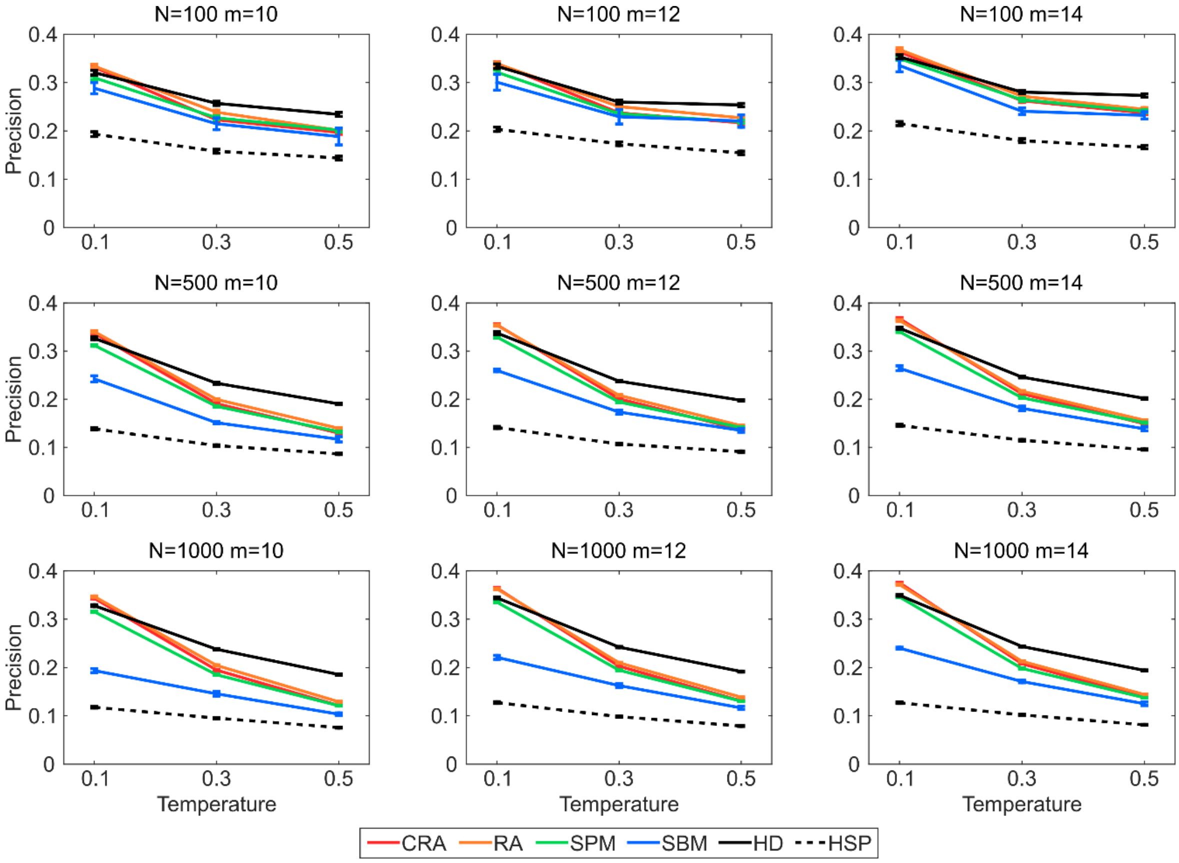


Figure 5. Link prediction on PSO networks.

Synthetic networks have been generated using the PSO model with parameters N = [100, 500, 1000], m = [10, 12, 14], T = [0.1, 0.3, 0.5] and γ = 3. For each combination of parameters, 100 networks have been generated (only 10 repetitions for SBM due to the high computational time). For each network 10% of links have been randomly removed and the algorithms have been executed in order to assign likelihood scores to the non-observed links in these reduced networks. In order to evaluate the performance, the links are ranked by likelihood scores and the precision is computed as the percentage of removed links among the top-r in the ranking, where r is the total number of links removed. The plots report for each parameter combination the mean precision and standard error over the random repetitions.

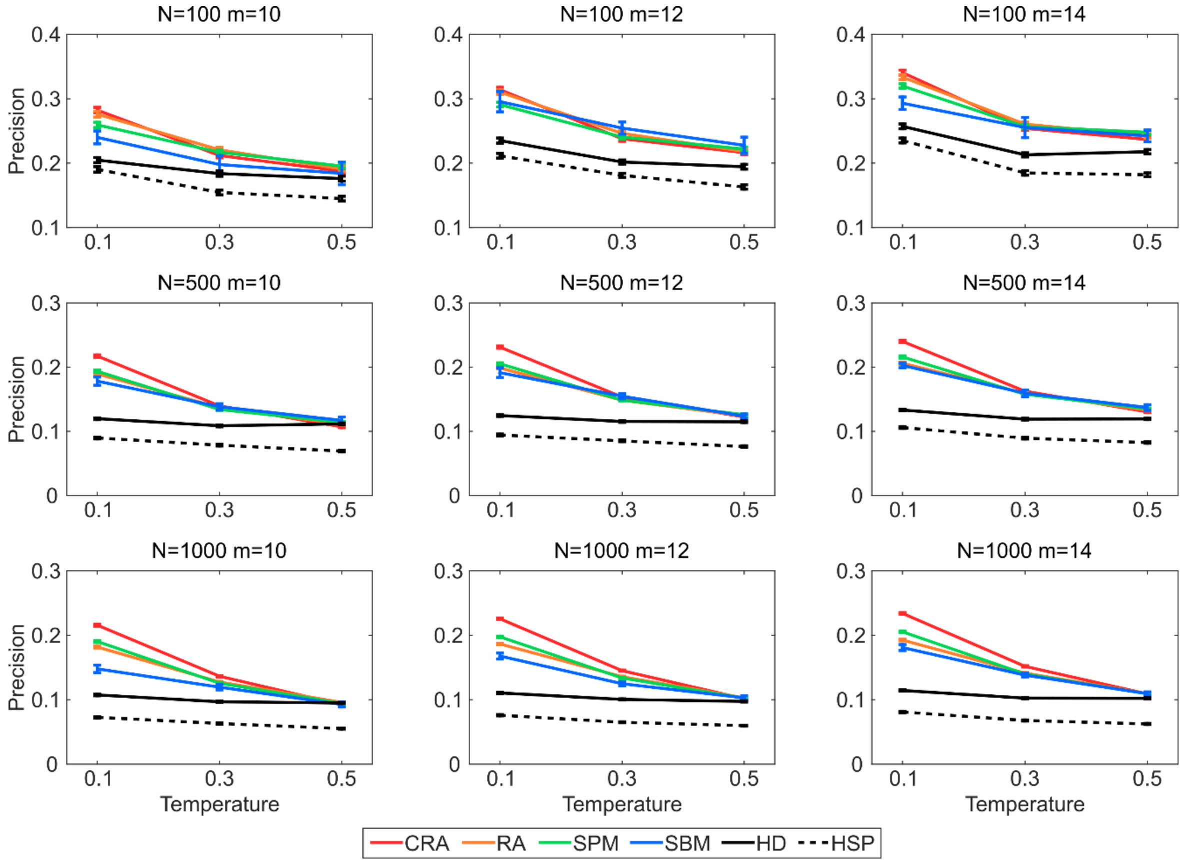


Figure 6. Link prediction on nPSO networks with 4 communities.
Synthetic networks have been generated using the nPSO model with parameters $N = [100, 500, 1000]$, $m = [10, 12, 14]$, $T = [0.1, 0.3, 0.5]$, $\gamma = 3$ and angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components $C = 4$. For each combination of parameters, 100 networks have been generated (only 10 repetitions for SBM due to the high computational time). For each network 10% of links have been randomly removed and the algorithms have been executed in order to assign likelihood scores to the non-observed links in these reduced networks. In order to evaluate the performance, the links are ranked by likelihood scores and the precision is computed as the percentage of removed links among the top-*r* in the ranking, where *r* is the total number of links removed. The plots report for each parameter combination the mean precision and standard error over the random repetitions.

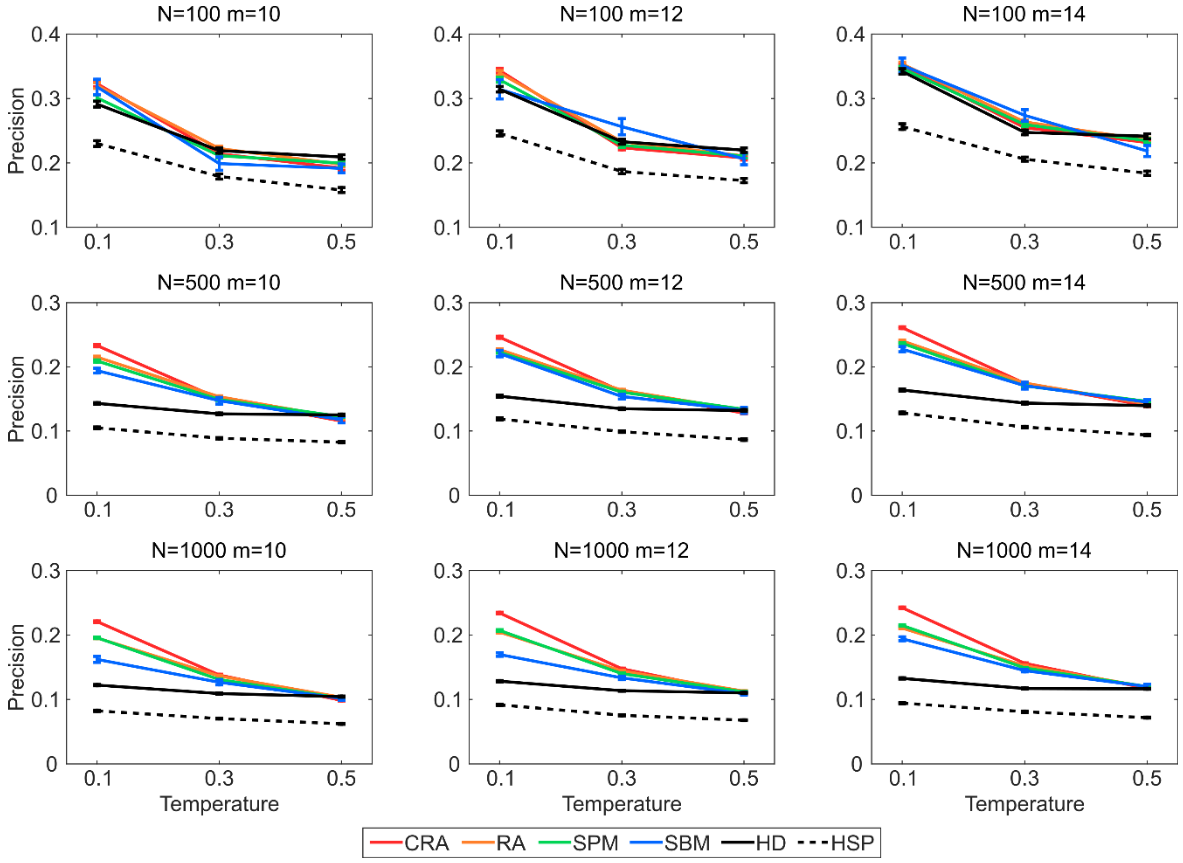


Figure 7. Link prediction on nPSO networks with 8 communities.
Synthetic networks have been generated using the nPSO model with parameters $N = [100, 500, 1000]$, $m = [10, 12, 14]$, $T = [0.1, 0.3, 0.5]$, $\gamma = 3$ and angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components $C = 8$. For each combination of parameters, 100 networks have been generated (only 10 repetitions for SBM due to the high computational time). For each network 10% of links have been randomly removed and the algorithms have been executed in order to assign likelihood scores to the non-observed links in these reduced networks. In order to evaluate the performance, the links are ranked by likelihood scores and the precision is computed as the percentage of removed links among the top-*r* in the ranking, where *r* is the total number of links removed. The plots report for each parameter combination the mean precision and standard error over the random repetitions.

Table 1. Community detection on nPSO networks using network embedding and Louvain. Synthetic networks have been generated using the nPSO model with parameters N = [100, 500, 1000], m = [10, 12, 14], T = [0.1, 0.3, 0.5], γ = 3 and angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components C = [4, 8]. For each combination of parameters, 10 networks have been generated. Each network has been embedded in the hyperbolic space using coalescent embedding techniques and the embedding coordinates are used to weight the input matrix for the Louvain algorithm: observed links are weighted using the hyperbolic distances (HD) between the nodes and, in a second variant, non-observed links using the hyperbolic shortest paths (HSP) (see Methods for details). As a reference, the Louvain algorithm has been run giving in input also the unweighted adjacency matrix. The communities detected have been compared to the annotated ones computing the Normalized Mutual Information (NMI). The table reports the mean NMI for nPSO networks with 4 and 8 communities, the mean NMI over all the networks and the mean ranking. The best coalescent embedding technique and the unweighted variant are highlighted.

	nPSO C=4	**nPSO C=8**	**mean NMI**	**mean ranking**
RA-LE-HD	**0.95**	**0.91**	**0.93**	**11.5**
EBC-LE-HSP	0.96	0.91	0.93	11.7
EBC-ncISO-HD	0.96	0.90	0.93	11.8
EBC-ISO-HD	0.96	0.89	0.93	11.9
EBC-LE-HD	0.95	0.91	0.93	11.9
RA-LE-HSP	0.96	0.91	0.93	12.0
RA-ISO-HD	0.95	0.89	0.92	13.8
RA-ncISO-HD	0.95	0.89	0.92	13.9
EBC-ncISO-HSP	0.95	0.89	0.92	15.9
RA-ncISO-EA-HD	0.95	0.87	0.91	16.1
RA-ISO-EA-HD	0.95	0.86	0.91	16.6
EBC-ISO-EA-HD	0.96	0.85	0.91	16.6
EBC-ISO-HSP	0.95	0.88	0.92	16.8
EBC-ncISO-EA-HD	0.95	0.85	0.90	17.4
RA-LE-EA-HD	0.94	0.86	0.90	17.7
RA-ncMCE-EA-HD	0.95	0.85	0.90	18.0
EBC-LE-EA-HD	0.95	0.85	0.90	18.2
RA-MCE-HD	0.93	0.86	0.90	18.3
RA-ncMCE-HD	0.92	0.88	0.90	18.5
EBC-MCE-HD	0.94	0.86	0.90	18.7
RA-MCE-EA-HD	0.94	0.85	0.90	18.9
EBC-ncMCE-HD	0.92	0.87	0.90	19.1
unweighted	**0.95**	**0.83**	**0.89**	**19.2**
RA-ncISO-HSP	0.94	0.85	0.89	20.7
EBC-ncMCE-EA-HD	0.93	0.83	0.88	21.3
RA-ISO-HSP	0.93	0.85	0.89	21.5
EBC-MCE-EA-HD	0.93	0.82	0.88	22.2
EBC-ncMCE-HSP	0.90	0.84	0.87	24.4
RA-ncISO-EA-HSP	0.90	0.81	0.86	27.4
EBC-ISO-EA-HSP	0.90	0.81	0.86	27.9
RA-ISO-EA-HSP	0.90	0.81	0.86	28.0
EBC-ncISO-EA-HSP	0.89	0.81	0.85	28.2
EBC-MCE-HSP	0.89	0.80	0.85	29.2
RA-LE-EA-HSP	0.88	0.81	0.85	29.4
RA-ncMCE-HSP	0.87	0.80	0.84	29.4
EBC-LE-EA-HSP	0.87	0.80	0.84	29.7
RA-ncMCE-EA-HSP	0.89	0.79	0.84	29.8
RA-MCE-HSP	0.87	0.79	0.83	30.4
EBC-ncMCE-EA-HSP	0.87	0.79	0.83	31.0
RA-MCE-EA-HSP	0.88	0.78	0.83	31.1
EBC-MCE-EA-HSP	0.83	0.76	0.79	34.6

Table 2. Community detection on nPSO networks using network embedding and Infomap. Synthetic networks have been generated using the nPSO model with parameters N = [100, 500, 1000], m = [10, 12, 14], T = [0.1, 0.3, 0.5], γ = 3 and angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components C = [4, 8]. For each combination of parameters, 10 networks have been generated. Each network has been embedded in the hyperbolic space using coalescent embedding techniques and the embedding coordinates are used to weight the input matrix for the Infomap algorithm: observed links are weighted using the hyperbolic distances (HD) between the nodes (see Methods for details). As a reference, the Infomap algorithm has been run giving in input also the unweighted adjacency matrix. The communities detected have been compared to the annotated ones computing the Normalized Mutual Information (NMI). The table reports the mean NMI for nPSO networks with 4 and 8 communities, the mean NMI over all the networks and the mean ranking. The best coalescent embedding technique and the unweighted variant are highlighted.

	nPSO C=4	nPSO C=8	mean NMI	mean ranking
EBC-ncISO-HD	**0.89**	**0.80**	**0.84**	**8.9**
EBC-ISO-HD	0.88	0.79	0.84	9.1
EBC-LE-HD	0.91	0.81	0.86	9.6
RA-LE-HD	0.92	0.81	0.87	10.2
EBC-ISO-EA-HD	0.81	0.73	0.77	10.5
EBC-ncISO-EA-HD	0.80	0.73	0.77	10.6
unweighted	**0.77**	**0.71**	**0.74**	**10.7**
RA-ISO-EA-HD	0.81	0.73	0.77	10.7
RA-ISO-HD	0.89	0.80	0.84	10.8
RA-ncISO-HD	0.89	0.79	0.84	11.0
EBC-LE-EA-HD	0.80	0.73	0.77	11.0
RA-ncISO-EA-HD	0.80	0.73	0.77	11.0
RA-LE-EA-HD	0.80	0.73	0.76	11.4
EBC-MCE-EA-HD	0.79	0.73	0.76	11.4
EBC-ncMCE-EA-HD	0.80	0.73	0.76	11.4
EBC-MCE-HD	0.83	0.77	0.80	11.5
RA-ncMCE-EA-HD	0.80	0.73	0.76	11.7
RA-MCE-EA-HD	0.79	0.73	0.76	12.0
EBC-ncMCE-HD	0.88	0.83	0.86	12.0
RA-ncMCE-HD	0.87	0.81	0.84	12.8
RA-MCE-HD	0.83	0.76	0.80	12.9

C.6 Supplementary Information

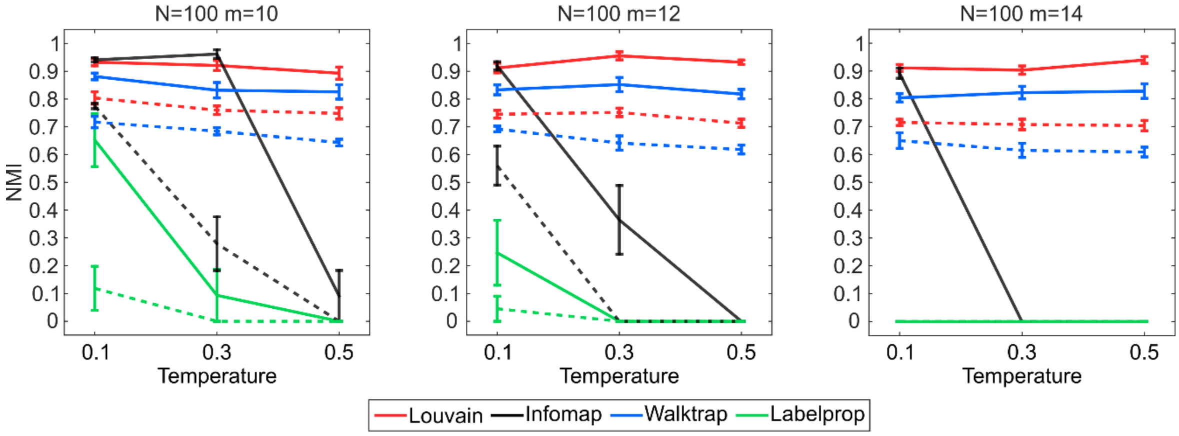


Figure 1. Comparison of performance on small nPSO networks for 4 and 8 communities. The same results reported in Figure 1-2 in the main article for networks with *N*=100 are here shown in a unique plot, in order to highlight the decrease of performance of the community detection methods when the number of communities is increased from 4 (full lines) to 8 (dashed lines).

Table 1. Community detection on nPSO networks using network embedding and Walktrap. Synthetic networks have been generated using the nPSO model with parameters N = [100, 500, 1000], m = [10, 12, 14], T = [0.1, 0.3, 0.5], γ = 3 and angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components C = [4, 8]. For each combination of parameters, 10 networks have been generated. Each network has been embedded in the hyperbolic space using coalescent embedding techniques and the embedding coordinates are used to weight the input matrix for the Walktrap algorithm: observed links are weighted using the hyperbolic distances (HD) between the nodes (see Methods for details). As a reference, the Walktrap algorithm has been run giving in input also the unweighted adjacency matrix. The communities detected have been compared to the annotated ones computing the Normalized Mutual Information (NMI). The table reports the mean NMI for nPSO networks with 4 and 8 communities, the mean NMI over all the networks and the mean ranking. The best coalescent embedding technique and the unweighted variant are highlighted.

	nPSO C=4	nPSO C=8	mean NMI	mean ranking
EBC-ISO-HD	**0.93**	**0.88**	**0.91**	**9.7**
EBC-ncISO-HD	0.93	0.88	0.90	9.8
RA-ISO-EA-HD	0.92	0.88	0.90	10.4
RA-LE-HD	0.92	0.88	0.90	10.5
EBC-LE-HD	0.92	0.88	0.90	10.6
EBC-ISO-EA-HD	0.92	0.88	0.90	10.6
RA-ncISO-EA-HD	0.92	0.88	0.90	10.6
EBC-ncISO-EA-HD	0.92	0.88	0.90	10.7
RA-ncISO-HD	0.91	0.88	0.90	10.8
EBC-ncMCE-EA-HD	0.92	0.88	0.90	10.9
RA-ISO-HD	0.91	0.88	0.90	10.9
EBC-ncMCE-HD	0.90	0.88	0.89	11.1
RA-LE-EA-HD	0.92	0.88	0.90	11.2
EBC-LE-EA-HD	0.92	0.88	0.90	11.4
EBC-MCE-EA-HD	0.92	0.87	0.89	11.5
RA-ncMCE-EA-HD	0.92	0.87	0.89	11.6
RA-MCE-EA-HD	0.92	0.87	0.89	11.6
RA-ncMCE-HD	0.90	0.88	0.89	11.7
EBC-MCE-HD	0.91	0.87	0.89	11.7
RA-MCE-HD	0.91	0.87	0.89	11.8
unweighted	**0.91**	**0.87**	**0.89**	**11.9**

Table 2. Community detection on nPSO networks using network embedding and Label propagation.

Synthetic networks have been generated using the nPSO model with parameters N = [100, 500, 1000], m = [10, 12, 14], T = [0.1, 0.3, 0.5], γ = 3 and angular coordinates sampled according to a Gaussian mixture distribution with equal proportions and components C = [4, 8]. For each combination of parameters, 10 networks have been generated. Each network has been embedded in the hyperbolic space using coalescent embedding techniques and the embedding coordinates are used to weight the input matrix for the Label propagation algorithm: observed links are weighted using the hyperbolic distances (HD) between the nodes (see Methods for details). As a reference, the Label propagation algorithm has been run giving in input also the unweighted adjacency matrix. The communities detected have been compared to the annotated ones computing the Normalized Mutual Information (NMI). The table reports the mean NMI for nPSO networks with 4 and 8 communities, the mean NMI over all the networks and the mean ranking. The best coalescent embedding technique and the unweighted variant are highlighted.

	nPSO C=4	**nPSO C=8**	**mean NMI**	**mean ranking**
EBC-LE-HD	**0.74**	**0.66**	**0.70**	**8.2**
RA-LE-HD	0.73	0.66	0.70	8.6
EBC-ISO-HD	0.69	0.63	0.66	9.1
EBC-ncISO-HD	0.67	0.63	0.65	9.6
RA-ISO-HD	0.69	0.62	0.65	9.9
EBC-ncMCE-HD	0.65	0.65	0.65	10.2
RA-ncISO-HD	0.68	0.61	0.64	10.3
unweighted	**0.57**	**0.56**	**0.56**	**10.8**
EBC-MCE-HD	0.58	0.60	0.59	11.0
RA-ISO-EA-HD	0.58	0.59	0.58	11.4
RA-LE-EA-HD	0.59	0.58	0.59	11.6
RA-ncISO-EA-HD	0.55	0.58	0.57	11.7
EBC-ISO-EA-HD	0.56	0.56	0.56	11.8
RA-ncMCE-EA-HD	0.56	0.58	0.57	11.8
EBC-ncISO-EA-HD	0.55	0.57	0.56	11.9
RA-ncMCE-HD	0.61	0.63	0.62	11.9
RA-MCE-HD	0.57	0.59	0.58	12.0
EBC-MCE-EA-HD	0.55	0.56	0.55	12.1
EBC-ncMCE-EA-HD	0.52	0.57	0.55	12.1
EBC-LE-EA-HD	0.54	0.56	0.55	12.4
RA-MCE-EA-HD	0.51	0.56	0.54	12.7

Table 3. Link prediction on small-size real networks.

For each network 10% of links have been randomly removed (10 repetitions for SBM due to the high computational time, 100 repetitions for the others) and the algorithms have been executed in order to assign likelihood scores to the non-observed links in these reduced networks. In order to evaluate the performance, the links are ranked by likelihood scores and the precision is computed as the percentage of removed links among the top-*r* in the ranking, where *r* is the total number of links removed. The table reports for each network the mean precision over the random repetitions. The last rows show the mean precision and the mean ranking over the entire dataset. For each network the best method is highlighted in bold. The networks are sorted by increasing number of nodes *N*.

	SPM	**CRA**	**SBM**	**RA**
mouse neural	0.02	**0.11**	0.10	0.07
karate	0.17	0.20	**0.28**	0.14
dolphins	0.13	0.14	**0.16**	0.10
macaque neural	**0.72**	0.56	0.68	0.51
polbooks	0.17	0.17	0.15	**0.21**
ACM2009 contacts	0.26	**0.27**	0.25	**0.27**
football	0.31	**0.36**	0.34	0.27
physicians innovation	0.07	0.07	0.06	**0.08**
manufacturing email	**0.51**	0.42	0.47	0.43
littlerock foodweb	**0.84**	0.15	0.73	0.10
jazz	**0.65**	0.56	0.47	0.54
residence hall friends	**0.28**	0.24	0.18	0.25
haggle contacts	**0.62**	0.57	**0.62**	0.58
worm nervoussys	**0.16**	0.12	0.15	0.10
netsci	0.41	0.50	0.13	**0.59**
infectious contacts	**0.37**	0.34	0.30	0.35
flightmap	**0.75**	0.54	0.64	0.56
email	**0.16**	**0.16**	0.09	0.15
polblog	**0.23**	0.17	0.19	0.13
mean precision	**0.36**	0.30	0.32	0.29
mean ranking	**1.89**	2.53	2.71	2.87

Table 4. Link prediction on large-size real networks.

For each network 10% of links have been randomly removed (10 repetitions) and the algorithms have been executed in order to assign likelihood scores to the non-observed links in these reduced networks. In order to evaluate the performance, the links are ranked by likelihood scores and the precision is computed as the percentage of removed links among the top-*r* in the ranking, where *r* is the total number of links removed. The table reports for each network the mean precision over the random repetitions. The last rows show the mean precision and the mean ranking over the entire dataset. For each network the best method is highlighted in bold. The networks are sorted by increasing number of nodes *N*.

	CRA	**RA**	**SPM**
odlis	**0.12**	0.10	0.08
advogato	**0.16**	0.14	0.15
arxiv astroph	0.53	0.64	**0.67**
thesaurus	**0.08**	0.03	0.07
arxiv hepth	0.22	0.20	**0.27**
ARK200909	**0.15**	0.14	0.10
ARK200912	**0.16**	0.15	0.09
ARK201003	**0.16**	0.15	0.10
ARK201006	**0.16**	0.15	0.10
ARK201009	**0.16**	**0.16**	0.10
ARK201012	**0.16**	**0.16**	0.11
facebook	**0.11**	0.06	0.10
mean precision	**0.18**	0.17	0.16
mean ranking	**1.33**	2.25	2.42

Table 5. Link prediction in time on AS Internet networks.

Six AS Internet network snapshots are available from September 2009 to December 2010, at time steps of 3 months. For every snapshot at times $i = [1, 5]$ the algorithms have been executed in order to assign likelihood scores to the non-observed links and the link prediction performance has been evaluated computing the precision with respect to every future time point $j = [i+1, 6]$. Considering a pair of time points (i, j), the non-observed links at time i are ranked by decreasing likelihood scores and the precision is computed as the percentage of links that appear at time j among the top-r in the ranking, where r is the total number of non-observed links at time i that appear at time j. Non-observed links at time i involving nodes that disappear at time j are not considered in the ranking. The table reports for each method a 5-dimensional upper triangular matrix, containing as element (i, j) the precision of the link prediction from time i to time j+1. On the bottom-right side, the methods are ranked by the mean precision computed over all the time combinations, the mean ranking is also shown. For each comparison the best method is highlighted in bold.

CRA					**RA**				
0.11	**0.12**	**0.13**	**0.14**	**0.14**	0.10	0.11	0.11	0.12	0.12
	0.12	**0.13**	**0.14**	**0.14**		0.09	0.10	0.11	0.12
		0.12	**0.13**	**0.14**			0.09	0.11	0.12
			0.12	**0.13**				0.10	0.11
				0.12					0.10
SPM							mean precision		mean ranking
0.08	0.09	0.09	0.10	0.11		**CRA**	**0.13**		**1**
	0.07	0.08	0.09	0.10		RA	0.11		2
		0.08	0.09	0.10		SPM	0.09		3
			0.08	0.09					
				0.09					

Table 6. Statistics of small-size real networks.

For each network several statistics have been computed. *N* is the number of nodes. *E* is the number of edges. The parameter *m*, as in the PSO model, refers to half of the average node degree. *D* is the network density. *C* is the average clustering coefficient, computed for each node as the number of links between its neighbours over the number of possible links [37]. *L* is the characteristic path length of the network [37]. Power-law is the exponent γ of the power-law distribution estimated from the observed degree distribution of the network using the maximum likelihood procedure described in [91].

	N	E	m	D	C	L	Power law
mouse neural	18	37	2.06	0.24	0.22	1.97	4.01
karate	34	78	2.29	0.14	0.57	2.41	2.12
dolphins	62	159	2.56	0.08	0.26	3.36	6.96
macaque neural	94	1515	16.12	0.35	0.77	1.77	4.46
polbooks	105	441	4.20	0.08	0.49	3.08	2.62
ACM2009 contacts	113	2196	19.43	0.35	0.53	1.66	3.74
football	115	613	5.33	0.09	0.40	2.51	9.09
physicians innovation	117	465	3.97	0.07	0.22	2.59	4.51
manufacturing email	167	3250	19.46	0.23	0.59	1.97	3.13
littlerock foodweb	183	2434	13.30	0.15	0.32	2.15	3.00
jazz	198	2742	13.85	0.14	0.62	2.24	4.48
residence hall friends	217	1839	8.47	0.08	0.36	2.39	6.32
haggle contacts	274	2124	7.75	0.06	0.63	2.42	1.51
worm nervoussys	297	2148	7.23	0.05	0.29	2.46	3.34
netsci	379	914	2.41	0.01	0.74	6.04	3.36
infectious contacts	410	2765	6.74	0.03	0.46	3.63	6.42
flightmap	456	37947	83.22	0.37	0.81	1.64	1.71
email	1133	5451	4.81	0.01	0.22	3.61	4.89
polblog	1222	16714	13.68	0.02	0.32	2.74	2.38

Table 7. Statistics of AS Internet snapshots and large-size real networks.

The first half of the table reports the statistics for the AS Internet snapshots, whereas the second half the large-size real networks. Note that also the last AS Internet snapshot has been considered in the simulations on the large-size real networks. For each network several statistics have been computed. *N* is the number of nodes. *E* is the number of edges. The parameter *m*, as in the PSO model, refers to half of the average node degree. *D* is the network density. *C* is the average clustering coefficient, computed for each node as the number of links between its neighbours over the number of possible links [37]. *L* is the characteristic path length of the network [37]. Power-law is the exponent γ of the power-law distribution estimated from the observed degree distribution of the network using the maximum likelihood procedure described in [91].

	N	E	m	D	C	L	Power law
ARK200909	24091	59531	2.47	0.0039	0.36	3.53	2.12
ARK200912	25910	63435	2.45	0.0031	0.36	3.54	2.11
ARK201003	26307	66089	2.51	0.0012	0.37	3.53	2.26
ARK201006	26756	68150	2.55	0.0011	0.37	3.51	2.08
ARK201009	28353	73722	2.60	0.0009	0.37	3.52	2.23
ARK201012	29333	78054	2.66	0.0002	0.38	3.50	2.22
odlis	2898	16376	5.65	0.0002	0.30	3.17	2.63
advogato	5042	39227	7.78	0.0002	0.25	3.27	2.73
arxiv astroph	17903	196972	11.00	0.0002	0.63	4.19	2.83
thesaurus	23132	297094	12.84	0.0002	0.09	3.49	2.84
arxiv hepth	27400	352021	12.85	0.0002	0.31	4.28	2.86
facebook	43953	182384	4.15	0.0002	0.11	5.60	3.66

Table 8. Link prediction on small-size real networks using hyperbolic embedding methods. For each of the small-size real networks shown in Table 1, 10% of links have been randomly removed (100 repetitions) and the algorithms have been executed in order to assign likelihood scores to the non-observed links in these reduced networks. In order to evaluate the performance, the links are ranked by likelihood scores and the precision is computed as the percentage of removed links among the top-*r* in the ranking, where *r* is the total number of links removed. The table reports the mean precision and the mean ranking over the entire dataset. The methods are sorted by mean ranking.

	mean precision	mean ranking
RA-ncMCE-HSP	0.17	10.8
EBC-ncISO-HSP	0.17	10.9
RA-ncISO-HSP	0.17	11.1
RA-MCE-EA-HSP	0.17	11.2
RA-MCE-HSP	0.17	11.8
EBC-ncISO-EA-HSP	0.16	12.8
RA-LE-HSP	0.17	12.9
RA-ISO-HSP	0.16	13.1
RA-ncMCE-EA-HSP	0.17	13.7
RA-ncISO-EA-HSP	0.16	13.8
EBC-ISO-HSP	0.15	13.9
EBC-ncMCE-HSP	0.16	14.0
EBC-MCE-HSP	0.16	14.2
EBC-ISO-EA-HSP	0.16	14.3
RA-LE-EA-HSP	0.16	14.8
EBC-ncMCE-EA-HSP	0.16	15.1
EBC-LE-EA-HSP	0.15	15.1
EBC-MCE-EA-HSP	0.16	15.6
RA-ISO-EA-HSP	0.16	16.2
EBC-LE-HSP	0.14	19.7
LPCS-HSP	0.14	20.4
EBC-ncISO-EA-HD	0.12	24.7
RA-LE-EA-HD	0.12	26.7
EBC-MCE-EA-HD	0.12	27.0
EBC-ncMCE-EA-HD	0.12	27.2
RA-ncMCE-EA-HD	0.12	27.7
RA-ncISO-EA-HD	0.12	27.7
LPCS-HD	0.12	27.8
HyperMapCN-HSP	0.11	27.9
EBC-ISO-EA-HD	0.11	28.3
RA-ISO-EA-HD	0.11	28.9
RA-ncISO-HD	0.10	29.3
RA-ncMCE-HD	0.10	29.4
EBC-LE-EA-HD	0.11	29.6
EBC-ncISO-HD	0.10	29.8
RA-MCE-EA-HD	0.11	30.4
RA-ISO-HD	0.09	31.8
RA-MCE-HD	0.09	32.6
EBC-ncMCE-HD	0.09	33.1
HyperMapCN-HD	0.06	33.2
RA-LE-HD	0.09	34.1
EBC-MCE-HD	0.09	34.3
EBC-ISO-HD	0.07	35.4
EBC-LE-HD	0.07	37.3

Table 9. Top-ranked intra-community links using hyperbolic embedding methods.

For three small-size real networks with community metadata available, 10% of links have been randomly removed (100 repetitions) and the algorithms have been executed in order to assign likelihood scores to the non-observed links in these reduced networks. After ranking the links by likelihood scores, the top-r in the ranking are considered, where r is the total number of links removed. The table reports the percentage of intra-community links in the top-ranking (icl_top/top) and the percentage of these top-ranked intra-community links corresponding to correct predictions (icl_top_true/icl_top), averaged over the 100 repetitions. The mean over all the techniques is also highlighted.

	Karate		**Polbooks**		**Polblog**	
	$\frac{icl\_top}{top}$	$\frac{icl\_top\_true}{icl\_top}$	$\frac{icl\_top}{top}$	$\frac{icl\_top\_true}{icl\_top}$	$\frac{icl\_top}{top}$	$\frac{icl\_top\_true}{icl\_top}$
EBC-ISO-HD	1.00	0.01	0.95	0.06	0.93	0.05
EBC-ISO-HSP	0.97	0.09	0.98	0.09	0.94	0.14
EBC-ISO-EA-HD	0.50	0.20	0.82	0.10	0.78	0.10
EBC-ISO-EA-HSP	0.91	0.07	0.94	0.09	0.76	0.13
EBC-ncISO-HD	1.00	0.02	0.92	0.08	0.97	0.05
EBC-ncISO-HSP	0.99	0.06	0.96	0.10	0.98	0.13
EBC-ncISO-EA-HD	0.48	0.24	0.83	0.11	0.87	0.11
EBC-ncISO-EA-HSP	0.73	0.07	0.93	0.10	0.84	0.12
EBC-MCE-HD	0.98	0.01	0.88	0.07	0.54	0.04
EBC-MCE-HSP	0.88	0.07	0.94	0.11	0.68	0.12
EBC-MCE-EA-HD	0.35	0.35	0.83	0.10	0.46	0.08
EBC-MCE-EA-HSP	0.67	0.07	0.94	0.10	0.72	0.08
EBC-ncMCE-HD	0.98	0.01	0.85	0.06	0.54	0.05
EBC-ncMCE-HSP	0.91	0.04	0.84	0.09	0.50	0.13
EBC-ncMCE-EA-HD	0.36	0.31	0.76	0.10	0.43	0.10
EBC-ncMCE-EA-HSP	0.79	0.07	0.84	0.10	0.59	0.09
EBC-LE-HD	1.00	0.02	0.90	0.07	0.44	0.04
EBC-LE-HSP	0.99	0.09	0.95	0.10	0.67	0.07
EBC-LE-EA-HD	0.44	0.24	0.84	0.10	0.41	0.06
EBC-LE-EA-HSP	0.78	0.10	0.93	0.11	0.67	0.08
RA2-ISO-HD	0.99	0.01	0.91	0.07	0.95	0.04
RA2-ISO-HSP	0.98	0.11	0.96	0.08	0.99	0.09
RA2-ISO-EA-HD	0.47	0.25	0.81	0.09	0.85	0.07
RA2-ISO-EA-HSP	0.83	0.09	0.94	0.08	0.90	0.08
RA2-ncISO-HD	0.99	0.02	0.93	0.07	0.94	0.06
RA2-ncISO-HSP	0.99	0.09	0.96	0.09	0.99	0.10
RA2-ncISO-EA-HD	0.39	0.29	0.82	0.09	0.87	0.10
RA2-ncISO-EA-HSP	0.80	0.10	0.93	0.08	0.83	0.10
RA2-MCE-HD	0.97	0.01	0.86	0.06	0.89	0.05
RA2-MCE-HSP	0.96	0.09	0.92	0.08	0.97	0.10
RA2-MCE-EA-HD	0.32	0.34	0.75	0.08	0.80	0.08
RA2-MCE-EA-HSP	0.74	0.12	0.88	0.08	0.93	0.09
RA2-ncMCE-HD	1.00	0.01	0.81	0.07	0.91	0.05
RA2-ncMCE-HSP	0.99	0.07	0.83	0.08	0.96	0.10
RA2-ncMCE-EA-HD	0.42	0.31	0.70	0.09	0.81	0.08
RA2-ncMCE-EA-HSP	0.82	0.08	0.87	0.07	0.87	0.09
RA2-LE-HD	1.00	0.02	0.91	0.07	0.96	0.03
RA2-LE-HSP	0.99	0.10	0.92	0.11	0.99	0.09
RA2-LE-EA-HD	0.46	0.28	0.81	0.10	0.92	0.09
RA2-LE-EA-HSP	0.87	0.09	0.93	0.09	0.94	0.09
HyperMapCN-HD	0.59	0.23	0.48	0.09	0.67	0.04
HyperMapCN-HSP	0.94	0.11	0.77	0.06	0.83	0.04
LPCS-HD	0.60	0.13	0.80	0.08	0.90	0.05
LPCS-HSP	0.74	0.12	0.88	0.09	0.92	0.08
mean	**0.78**	**0.12**	**0.87**	**0.09**	**0.80**	**0.08**

Bibliography

[1] E. Estrada, *The structure of complex network: Theory and applications* . 2011.

[2] L. Euler, “Solutio Problematis ad Geometriam Situs Pertinentis,” *Comment. Acad. Sci. U. Petrop*, vol. 8, pp. 128–140, 1736.

[3] A.-L. Barabási and M. Pósfai, *Network science*. Cambridge: Cambridge University Press, 2016.

[4] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks.,” *Nature*, vol. 393, no. 6684, pp. 440–2, 1998.

[5] S. Wasserman and K. Faust, “Social Network Analysis: Methods and Applications, Structural Analysis in the Social Sciences,” *Cambridge Univ. Press*, 1994.

[6] M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” *PNAS*, vol. 99, no. 12, pp. 7821–7826, 2002.

[7] M. E. J. Newman, “Fast algorithm for detecting community structure in networks,” *Phys. Rev. E*, vol. 69, no. 6, p. 66133, Jun. 2004.

[8] P. Erdös and A. Rényi, “On random graphs,” *Publ. Math.*, 1959.

[9] E. N. Gilbert, “Random Graphs,” *Ann. Math. Stat.*, vol. 30, no. 4, pp. 1141–1144, 1959.

[10] S. Milgram, “The Small World Problem,” *Psychol. Today*, 1967.

[11] A. L. Barabasi and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[12] A. D. Broido and A. Clauset, “Scale-free networks are rare,” *Nat. Commun.*, 2019.

[13] P. Holme, “Rare and everywhere: Perspectives on scale-free networks,” *Nature Communications*. 2019.

[14] I. Voitalov, P. van der Hoorn, R. van der Hofstad, and D. Krioukov, “Scale-free Networks Well Done,” *arXiv:1811.02071 [physics.soc-ph]*, 2018.

[15] M. Á. Serrano, D. Krioukov, and M. Boguñá, “Self-similarity of complex networks and hidden metric spaces,” *Phys. Rev. Lett.*, vol. 100, no. 7, pp. 1–4, 2008.

[16] M. Boguñá, D. Krioukov, and K. C. Claffy, “Navigability of complex networks,” *Nat. Phys.*, vol. 5, no. 1, pp. 74–80, 2008.

[17] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguñá, “Hyperbolic geometry of complex networks,” *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 82, no. 3, p. 036106, 2010.

[18] M. Boguñá, F. Papadopoulos, and D. Krioukov, “Sustaining the Internet with Hyperbolic Mapping,” *Nat. Commun.*, vol. 1, no. 6, pp. 1–8, 2010.

[19] D. Krioukov, F. Papadopoulos, A. Vahdat, and M. Boguñá, “Curvature and temperature of complex networks,” *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 80, no. 3, 2009.

[20] F. Papadopoulos, M. Kitsak, M. A. Serrano, M. Boguñá, and D. Krioukov, “Popularity versus similarity in growing networks,” *Nature*, vol. 489, no. 7417, pp. 537–540, 2012.

[21] M. McPherson, L. Smith-Lovin, and J. M. Cook, “Birds of a Feather: Homophily in Social Networks,” *Annu. Rev. Sociol.*, 2002.

[22] G. B. a.-L. Barabási, “Competition and multiscaling in evolving networks,” *Europhys. Lett.*, vol. 54, no. May, p. 13, 2000.

[23] F. Papadopoulos, C. Psomas, and D. Krioukov, “Network mapping by replaying hyperbolic growth,” *IEEE/ACM Trans. Netw.*, vol. 23, no. 1, pp. 198–211, 2015.

[24] Y. Yang, R. N. Lichtenwalter, N. V Chawla, Y. Yang, R. N. Lichtenwalter, and N. V Chawla, “Evaluating link prediction methods,” *Knowl. Inf. Syst.*, vol. 45, pp. 751–782, 2015.

[25] F. Papadopoulos, R. Aldecoa, and D. Krioukov, “Network Geometry Inference using Common Neighbors,” *Phys. Rev. E*, vol. 92, no. 2, p. 022807, 2015.

[26] G. Mangioni and A. Lima, “A growing model for scale-free networks embedded in hyperbolic metric spaces,” in *Studies in Computational Intelligence*, 2013.

[27] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, “A survey and comparison of peer-to-peer overlay network schemes,” *IEEE Communications Surveys and Tutorials*. 2005.

[28] K. Zuev, M. Boguñá, G. Bianconi, and D. Krioukov, “Emergence of Soft Communities from Geometric Preferential Attachment,” *Sci. Rep.*, vol. 5, p. 9421, 2015.

[29] D. Krioukov, “Clustering Implies Geometry in Networks,” *Phys. Rev. Lett.*, vol. 116, no. 20, pp. 1–5, 2016.

[30] K.-K. Kleineberg, M. Boguñá, M. Ángeles Serrano, and F. Papadopoulos, “Hidden geometric correlations in real multiplex networks,” *Nat. Phys.*, vol. 12, no. November, p. DOI: 10.1038/NPHYS3812, 2016.

[31] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, “Multilayer networks,” *J. Complex Networks*, vol. 2, no. 3, pp. 203–271, 2014.

[32] A. Allard, M. Á. Serrano, G. García-Pérez, and M. Boguñá, “The hidden geometry of weighted complex networks,” no. May 2016, pp. 1–8, 2016.

[33] M. Kitsak, I. Voitalov, and D. Krioukov, “Link prediction with hyperbolic geometry,” *arXiv:1903.08810 [physics.soc-ph]*, 2019.

[34] C. V. Cannistraci, T. Ravasi, F. M. Montevecchi, T. Ideker, and M. Alessio, “Nonlinear dimension reduction and clustering by Minimum Curvilinearity unfold neuropathic pain and tissue embryological classes,” *Bioinformatics*, vol. 26, pp. i531–i539, 2010.

[35] O. Kuchaiev, M. Rašajski, D. J. Higham, and N. Pržulj, “Geometric de-noising of protein-protein interaction networks,” *PLoS Comput. Biol.*, vol. 5, no. 8, p. e1000454, 2009.

[36] C. V. Cannistraci, G. Alanis-Lobato, and T. Ravasi, “Minimum curvilinearity to enhance topological prediction of protein interactions by network embedding,” *Bioinformatics*, vol. 29, no. 13, pp. 199–209, 2013.

[37] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks,” *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[38] A.-L. Barabasi, “Scale-Free Networks: A Decade and Beyond,” *Science*, vol. 325, no. 5939, pp. 412–413, 2009.

[39] M. E. J. Newman, “Modularity and community structure in networks.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 103, no. 23, pp. 8577–8582, 2006.

[40] S. Fortunato and D. Hric, “Community detection in networks: A user guide,” *Phys. Rep.*, vol. 659, pp. 1–44, 2016.

[41] L. Danon, A. Diaz-Guilera, and A. Arenas, “Effect of size heterogeneity on community identification in complex networks,” *J. Stat. Mech. Theory Exp.*, vol. 11010, no. 1, p. 6, 2006.

[42] A. Lancichinetti, S. Fortunato, and F. Radicchi, “Benchmark graphs for testing community detection algorithms,” *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 78, no. 4, 2008.

[43] B. Ball, B. Karrer, and M. E. J. Newman, “Efficient and principled method for detecting communities in networks,” *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 84, no. 3, 2011.

[44] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *J. Stat. Mech Theory Exp.*, vol. 2008, no. 10, p. 10008, 2008.

[45] M. Rosvall and C. T. Bergstrom, “Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems,” *PLoS One*, vol. 6, no. 4, p. e18209, 2011.

[46] P. Pons and M. Latapy, “Computing communities in large networks using random walks,” *J. Graph Algorithms Appl.*, vol. 10, no. 2, pp. 191–218, 2006.

[47] U. N. Raghavan, R. Albert, and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 76, no. 3, p. 036106, 2007.

[48] Z. Yang, R. Algesheimer, and C. J. Tessone, “A Comparative Analysis of Community Detection Algorithms on Artificial Networks,” *Sci. Rep.*, vol. 6, p. 30750, 2016.

[49] G. K. Orman and V. Labatut, “A Comparison of Community Detection Algorithms on Artificial Networks,” in *Discovery science*, 2009, pp. 242–256.

[50] A. Lancichinetti and S. Fortunato, “Community detection algorithms: A comparative analysis,” *Phys. Rev. E*, vol. 80, no. 5, p. 056117, 2009.

[51] A. Muscoloni, J. M. Thomas, S. Ciucci, G. Bianconi, and C. V. Cannistraci, “Machine learning meets complex networks via coalescent embedding in the hyperbolic space,” *Nat. Commun.*, vol. 8, 2017.

[52] R. Pech, D. Hao, L. Pan, H. Cheng, and T. Zhou, “Link Prediction via Matrix Completion,” *EPL*, no. 117, p. 38002, 2017.

[53] W. Wang, F. Cai, P. Jiao, and L. Pan, “A perturbation-based framework for link prediction via non-negative matrix factorization,” *Sci. Rep.*, vol. 6, no. December, p. 38938, 2016.

[54] C. V. Cannistraci, G. Alanis-Lobato, and T. Ravasi, “From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks,” *Sci. Rep.*, vol. 3, no. 1613, pp. 1–13, 2013.

[55] S. Daminelli, J. M. Thomas, C. Durán, and C. V. Cannistraci, “Common neighbours and the local-community-paradigm for topological link prediction in bipartite networks,” *New J. Phys.*, vol. 17, no. 11, p. 113037, 2015.

[56] C. Durán, S. Daminelli, J. M. Thomas, V. J. Haupt, M. Schroeder, and C. V. Cannistraci, “Pioneering topological methods for network-based drug–target prediction by exploiting a brain-network self-organization theory,” *Brief. Bioinform.*, vol. 8, no. W1, pp. 3–62, 2017.

[57] T. Zhou, L. Lu, and Y. C. Zhang, “Predicting missing links via local information,” *Eur. Phys. J. B*, vol. 71, no. 4, pp. 623–630, 2009.

[58] L. Lü, L. Pan, T. Zhou, Y.-C. Zhang, and H. E. Stanley, “Toward link predictability of complex networks,” *Proc. Natl. Acad. Sci.*, vol. 112, no. 8, pp. 2325–2330, 2015.

[59] R. Guimerà and M. Sales-Pardo, “Missing and spurious interactions and the reconstruction of complex networks,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 106, no. 52, pp. 22073–22078, 2009.

[60] G. García-Pérez, A. Allard, M. Á. Serrano, and M. Boguñá, “Mercator: uncovering faithful hyperbolic embeddings of complex networks,” *arXiv:1904.10814 [physics.soc-ph]*, 2019.

[61] G. Alanis-Lobato, P. Mier, and M. A. Andrade-Navarro, “Manifold learning and maximum likelihood estimation for hyperbolic network embedding,” *Appl. Netw. Sci.*, vol. 1, no. 1, p. 10, 2016.

[62] M. Keller-Ressel and S. Nargang, “Hydra: A method for strain-minimizing hyperbolic embedding,” *arXiv:1903.08977 [stat.CO]*, 2019.

[63] T. Blasius, T. Friedrich, A. Krohmer, and S. Laue, “Efficient Embedding of Scale-Free Graphs in the Hyperbolic Plane,” *IEEE/ACM Trans. Netw.*, 2018.

[64] K. Bringmann, R. Keusch, and J. Lengler, “Geometric inhomogeneous random graphs,” *Theor. Comput. Sci.*, vol. 760, pp. 35–54, 2019.

[65] A. Muscoloni and C. V. Cannistraci, “Minimum curvilinear automata with similarity attachment for network embedding and link prediction in the hyperbolic space,” *arXiv:1802.01183 [physics.soc-ph]*, 2018.

[66] R. C. Prim, “Shortest Connection Networks And Some Generalizations,” *Bell Syst. Tech. J.*, vol. 36, no. 6, pp. 1389–1401, 1957.

[67] A. Muscoloni and C. V. Cannistraci, “Navigability evaluation of complex networks by greedy routing efficiency,” *Proc. Natl. Acad. Sci.*, vol. 116, no. 5, pp. 1468 LP – 1469, Jan. 2019.

[68] A. Cacciola *et al.*, “Coalescent embedding in the hyperbolic space unsupervisedly discloses the hidden geometry of the brain,” *arXiv:1705.04192 [q-bio.NC]*, 2017.

[69] S. Mori, B. J. Crain, V. P. Chacko, and P. C. M. Van Zijl, “Three-dimensional tracking of axonal projections in the brain by magnetic resonance imaging,” *Ann. Neurol.*, 1999.

[70] D. J. Higham, M. Rašajski, and N. Pržulj, “Fitting a geometric graph to a protein-protein interaction network,” *Bioinformatics*, vol. 24, no. 8, pp. 1093–1099, 2008.

[71] Z. Wu, G. Menichetti, C. Rahmede, and G. Bianconi, “Emergent complex network geometry,” *Sci. Rep.*, vol. 5, p. 10073, Jan. 2015.

[72] G. Bianconi and C. Rahmede, “Complex Quantum Network Manifolds in Dimension d > 2 are Scale-Free.,” *Sci. Rep.*, vol. 5, p. 13979, Jan. 2015.

[73] G. Bianconi, “Interdisciplinary and physics challenges of network theory,” *Europhys. Lett.*, vol. 111, no. 5, p. 56001, Sep. 2015.

[74] G. Bianconi and C. Rahmede, “Emergent Hyperbolic Network Geometry,” *Sci. Rep.*, vol. 7, p. 41974, Feb. 2017.

[75] M. von Looz, M. Özdayi, S. Laue, and H. Meyerhenke, "Generating massive complex networks with hyperbolic geometry faster in practice," *arXiv:1606.09481 [cs.DS]*, 2016.

[76] T. Aste, T. Di Matteo, and S. T. Hyde, "Complex networks on hyperbolic surfaces," *Physica A*, vol. 346, pp. 20–26, 2005.

[77] W. S. Kennedy, O. Narayan, and I. Saniee, "On the Hyperbolicity of Large-Scale Networks," *arXiv:1307.0031 [physics.soc-ph]*, 2013.

[78] M. Borassi, A. Chessa, and G. Caldarelli, "Hyperbolicity measures democracy in real-world networks," *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.*, vol. 92, no. 3, p. 032812, 2015.

[79] V. Chepoi, F. F. Dragan, and Y. Vaxès, "Core congestion is inherent in hyperbolic networks," *arXiv:1605.03059 [cs.DS]*, 2016.

[80] E. A. Jonckheere, M. Lou, F. Bonahon, and Y. Baryshnikov, "Euclidean versus hyperbolic congestion in idealized versus experimental networks," *arXiv:0911.2538 [cs.NI]*, 2009.

[81] A. Muscoloni and C. V. Cannistraci, "Local-ring network automata and the impact of hyperbolic geometry in complex network link-prediction," *arXiv:1707.09496 [physics.soc-ph]*, 2017.

[82] E. Candellero and N. Fountoulakis, "Clustering and the hyperbolic geometry of complex networks," *Lect. Notes Comput. Sci.*, vol. 8882, pp. 1–12, 2014.

[83] C. Himpe and M. Ohlberger, "Model reduction for complex hyperbolic networks," *ECC 2014*, pp. 2739–2743, 2014.

[84] A. Muscoloni and C. V. Cannistraci, "A nonuniform popularity-similarity optimization (nPSO) model to efficiently generate realistic complex networks with communities," *arXiv:1707.07325 [physics.soc-ph]*, 2017.

[85] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction.," *Science*, vol. 290, pp. 2319–23, 2000.

[86] M. Belkin and P. Niyogi, "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering," *NIPS*, vol. 14, pp. 585–591, 2001.

[87] Z. Wang, Y. Wu, Q. Li, F. Jin, and W. Xiong, "Link prediction based on hyperbolic mapping with community structure for complex networks," *Physica A*, vol. 450, pp. 609–623, 2016.

[88] W. W. Zachary, "An Information Flow Model for Conflict and Fission in Small Groups," *J. Anthropol. Res.*, vol. 33, no. 4, pp. 452–473, 1977.

[89] W. Liu, X. Jiang, M. Pellegrini, and X. Wang, "Discovering communities in complex

networks by edge label propagation," *Sci. Rep.*, vol. 6, p. 22470, 2016.

[90] R. Cross and A. Parker, *The Hidden Power of Social Networks*, no. October. 2004.

[91] A. Clauset, C. Rohilla Shalizi, and M. E. J. Newman, "Power-Law Distributions in Empirical Data," *SIAM Rev.*, vol. 51, no. 4, pp. 661–703, 2009.

[92] S. T. Roweis and L. K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[93] M. Penrose, *Random geometric graphs*, no. 5. Oxford University Press, 2003.

[94] M. Gromov, *Metric structures for Riemannian and non-Riemannian spaces*. Springer Science & Business Media, 2007.

[95] L. Zagar, F. Mulas, S. Garagna, M. Zuccotti, R. Bellazzi, and B. Zupan, "Stage prediction of embryonic stem cell differentiation from genome-wide expression data," *Bioinformatics*, vol. 27, no. 18, pp. 2546–2553, 2011.

[96] G. Csardi and T. Nepusz, "The igraph software package for complex network research," *InterJournal*, vol. Complex Sy, p. 1695, 2006.

[97] D. A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," *Proc. IRE*, vol. 40, no. 9, pp. 1098–1101, 1952.

[98] L. Danon, A. Díaz-Guilera, J. Duch, A. Diaz-Guilera, and A. Arenas, "Comparing community structure identification," *J. Stat. Mech. Theory Exp.*, vol. 2005, no. 9, p. P09008, 2005.

[99] N. X. Vinh, J. Epps, and J. Bailey, "Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance," *J. Mach. Learn. Res.*, vol. 11, pp. 2837–2854, Dec. 2010.

[100] L. A. Adamic and N. Glance, "The Political Blogosphere and the 2004 U.S. Election: Divided They Blog," *LinkKDD 2005*, pp. 36–43, 2005.

[101] K. Claffy, Y. Hyun, K. Keys, M. Fomenkov, and D. Krioukov, "Internet mapping: From art to science," *CATCH 2009*, pp. 205–211, 2009.

[102] U. Brandes, "A faster algorithm for betweenness centrality," *J. Math. Sociol.*, vol. 25, no. 2, pp. 163–177, 2001.

[103] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Johns Hopkins University Press, 2013.

[104] P. Clemens, *Finite and Boundary Element Tearing and Interconnecting Solvers for Multiscale Problems*, Lecture No. Springer, 2013.

[105] R. B. Lehoucq and D. C. Sorensen, "Deflation Techniques for an Implicitly Restarted Arnoldi Iteration," *SIAM J. Matrix Anal. Appl.*, vol. 17, no. 4, pp. 789–821, 1996.

[106] R. Lehoucq, D. Sorensen, and C. Yang, *ARPACK Users' Guide*. Society for Industrial and Applied Mathematics, 1998.

[107] D. B. Johnson, "Efficient Algorithms for Shortest Paths in Sparse Networks," *J. ACM*, vol. 24, no. 1, pp. 1–13, 1977.

[108] R. M. Larsen, "Lanczos bidiagonalization with partial reorthogonalization," *DAIMI Rep. Ser.*, vol. 27, no. 537, 1998.

[109] J. B. Kruskal, "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem," *Proc. Am. Math. Soc.*, vol. 7, pp. 48–50, 1956.

[110] A. Muscoloni and C. V. Cannistraci, "Rich-clubness test: how to determine whether a complex network has or doesn't have a rich-club?," *arXiv:1704.03526v1 [physics.soc-ph]*, 2017.

[111] L. K. L. Saul and S. S. T. Roweis, "Think globally, fit locally: unsupervised learning of low dimensional manifolds," *J. Mach. Learn. Res.*, vol. 4, no. 1999, pp. 119–155, 2003.

[112] S. Zhou and R. J. Mondragón, "The Rich-Club Phenomenon in the Internet Topology," *IEEE Commun. Lett.*, vol. 8, no. 3, pp. 180–182, 2004.

[113] G. McLachlan and D. Peel, *Finite Mixture Models*. John Wiley & Sons, Inc., 2000.

[114] D. Hric, R. K. Darst, and S. Fortunato, "Community detection in networks: Structural communities versus ground truth," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 90, no. 6, 2014.

[115] A. Muscoloni and C. V. Cannistraci, "Leveraging the nonuniform PSO network model as a benchmark for performance evaluation in community detection and link prediction," *New J. Phys.*, vol. 20, 2018.

[116] M. E. J. Newman, "Assortative Mixing in Networks," *Phys. Rev. Lett.*, vol. 89, no. 20, 2002.

[117] Q. K. Telesford, K. E. Joyce, S. Hayasaka, J. H. Burdette, and P. J. Laurienti, "The ubiquity of small-world networks.," *Brain Connect.*, vol. 1, no. 5, pp. 367–75, 2011.

[118] V. Narula, A. G. Zippo, A. Muscoloni, G. E. M. Biella, and C. V. Cannistraci, "Can local-community-paradigm and epitopological learning enhance our understanding of how local brain connectivity is able to process, learn and memorize chronic pain?," *Appl. Netw. Sci.*, vol. 2, no. 1, p. 28, Aug. 2017.

[119] B. Karrer and M. E. J. Newman, "Stochastic blockmodels and community structure in networks," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 83, no. 1, 2011.

[120] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Phys. Rev.*

E - Stat. Nonlinear, Soft Matter Phys., vol. 80, no. 1, 2009.

[121] A. Muscoloni and C. V. Cannistraci, "A nonuniform popularity-similarity optimization (nPSO) model to efficiently generate realistic complex networks with communities," *New J. Phys.*, vol. 20, 2018.

[122] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *J. Stat. Mech. Theory Exp.*, vol. P09008, pp. 1–10, 2005.

[123] J. M. Thomas, A. Muscoloni, S. Ciucci, G. Bianconi, and C. V. Cannistraci, "Machine learning meets network science: dimensionality reduction for fast and efficient embedding of networks in the hyperbolic space," *arXiv:1602.06522*, 2016.

[124] Z. Wang, Q. Li, F. Jin, W. Xiong, and Y. Wu, "Hyperbolic mapping of complex networks based on community information," *Phys. A Stat. Mech. its Appl.*, vol. 455, pp. 1–29, 2016.

[125] D. D. Bock *et al.*, "Network anatomy and in vivo physiology of visual cortical neurons.," *Nature*, vol. 471, no. 7337, pp. 177–182, 2011.

[126] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations: Can geographic isolation explain this unique trait?," *Behav. Ecol. Sociobiol.*, vol. 54, no. 4, pp. 396–405, 2003.

[127] R. Kötter, "Online Retrieval, Processing, and Visualization of Primate Connectivity Data From the CoCoMac Database," *Neuroinformatics*, vol. 2, no. 2, pp. 127–144, 2004.

[128] L. Isella, J. Stehlé, A. Barrat, C. Cattuto, J.-F. Pinton, and W. Van den Broeck, "What's in a crowd? Analysis of face-to-face behavioral networks," *J. Theor. Biol.*, vol. 271, no. 1, pp. 166–180, 2011.

[129] J. Coleman, E. Katz, and H. Menzel, "The diffusion of an innovation among physicians," *Sociometry*, vol. 20, no. 4, pp. 253–270, 1957.

[130] R. Michalski, S. Palus, and P. Kazienko, "Matching Organizational Structure and Social Network Extracted from Email Communication," *Bus. Inf. Syst.*, vol. 87, pp. 197–206, 2011.

[131] N. D. Martinez, "Artifacts or attributes? Effects of resolution on the Little Rock Lake food web," *Ecol. Monogr.*, vol. 61, no. 4, pp. 367–392, 1991.

[132] P. M. Geiser and L. Danon, "Community structure in jazz," *Adv. Complex Syst.*, vol. 6, no. 4, pp. 565–573, 2003.

[133] L. C. Freeman, C. M. Webster, and D. M. Kirke, "Exploring social structure using dynamic three-dimensional color images," *Soc. Networks*, vol. 20, no. 2, pp. 109–118,

1998.

[134] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, “Impact of human mobility on opportunistic forwarding algorithms,” in *IEEE Transactions on Mobile Computing*, 2007, vol. 6, no. 6, pp. 606–620.

[135] M. E. J. Newman, “Finding community structure in networks using the eigenvectors of matrices,” *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 74, no. 3, 2006.

[136] B. J. Frey and D. Dueck, “Clustering by Passing Messages Between Data Points,” *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

[137] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, “Self-similar community structure in a network of human interactions.,” *Phys. Rev. E - Stat. Nonlinear Soft Matter Phys.*, vol. 68, no. 6, pp. 1–4, 2003.

[138] J. M. Reitz, *Online Dictionary for Library and Information Science*. 2002.

[139] P. Massa, M. Salvetti, and D. Tomasoni, “Bowling alone and trust decline in social network sites,” in *8th IEEE International Symposium on Dependable, Autonomic and Secure Computing, DASC 2009*, 2009, pp. 658–663.

[140] J. Leskovec, J. Kleinberg, and C. Faloutsos, “Graph evolution: Densification and Shrinking Diameters,” *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 1, 2007.

[141] G. R. Kiss, C. Armstrong, R. Milroy, and J. Piper, “An associative thesaurus of English and its computer analysis,” in *The computer and literary studies*, A. J. Aitkin, R. W. Bailey, and N. Hamilton-Smith, Eds. Edinburgh, UK: University Press, 1973.

[142] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, “On the Evolution of User Interaction in Facebook,” *Proc. 2nd ACM Work. Online Soc. networks - WOSN ’09*, p. 37, 2009.