Title
Bootstrapping on Undirected Binary Networks Via Statistical Mechanics

Permalink
https://escholarship.org/uc/item/0jq8h0f3

Journal
Journal of Statistical Physics, 156(5)

ISSN
0022-4715

Authors
Fushing, Hsieh
Chen, Chen
Liu, Shan-Yu
et al.

Publication Date
2014-09-01

DOI
10.1007/s10955-014-1043-6

Peer reviewed

Published in final edited form as:
J Stat Phys. 2014 September 1; 156(5): 823–842. doi:10.1007/s10955-014-1043-6.

Bootstrapping on undirected binary networks via statistical mechanics

Hsieh Fushing,
Department of Statistics, University of California, Davis, 1 Shields Ave, Davis, CA 95616
fhsieh@ucdavis.edu

Chen Chen,
Department of Statistics, University of California, Davis, 1 Shields Ave, Davis, CA 95616
chchen@ucdavis.edu

Shan-Yu Liu, and
Department of Statistics, University of California, Davis, 1 Shields Ave, Davis, CA 95616
shyliu@ucdavis.edu

Patrice Koehl
Department of Computer Science, University of California, Davis, 1 Shields Ave, Davis, CA 95616
koehl@cs.ucdavis.edu

Abstract

We propose a new method inspired from statistical mechanics for extracting geometric information from undirected binary networks and generating random networks that conform to this geometry. In this method an undirected binary network is perceived as a thermodynamic system with a collection of permuted adjacency matrices as its states. The task of extracting information from the network is then reformulated as a discrete combinatorial optimization problem of searching for its ground state. To solve this problem, we apply multiple ensembles of temperature regulated Markov chains to establish an ultrametric geometry on the network. This geometry is equipped with a tree hierarchy that captures the multiscale community structure of the network. We translate this geometry into a Parisi adjacency matrix, which has a relative low energy level and is in the vicinity of the ground state. The Parisi adjacency matrix is then further optimized by making block permutations subject to the ultrametric geometry. The optimal matrix corresponds to the macrostate of the original network. An ensemble of random networks is then generated such that each of these networks conforms to this macrostate; the corresponding algorithm also provides an estimate of the size of this ensemble. By repeating this procedure at different scales of the ultrametric geometry of the network, it is possible to compute its evolution entropy, i.e. to estimate the evolution of its complexity as we move from a coarse to a fine description of its geometric structure. We demonstrate the performance of this method on simulated as well as real data networks.

PACS 02.50.-r · 02.50.F · 02.50.Ga

Keywords

Bootstrapping; binary network; Parisi matrix

1 Introduction

A network is a set of objects (also called vertices, or nodes), with connections between them, also called edges [2,44,54,49,25,48]. As such, it is a mathematical construct that models the relationships between objects, where relationships look at the connections between one object and the rest of the network or more generally between subsets of objects, also referred to as clusters or communities. Networks are therefore tools that can capture interconnections between objects and allow for the analysis of these connections: they have become ubiquitous in many areas of science, with different definitions of nodes and edges. In biology for example, the objects (nodes) can be genes, proteins, neurons, individual organisms, species, etc. and edges can represent regulatory interactions, binding affinities, synapses, social associations, predation, gene flow and so on (see for examples [54,32,58,6,7]). Applications of network analysis also include friendship networks, the World Wide Web, disease models, among others [48,27].

Formal statistical models for the analysis of network data have emerged as a major research interests (see recent review articles by Airoldi and colleagues [25] and by Kolacyzk [30]). There are two main types of such statistical models: the static network models that concentrate on explaining the observed relationships based on a single version of the network, and the dynamic network models that are concerned with the mechanisms that govern changes in the network over time. We focus on the former, namely on untangling the complicated and convoluted relational patterns contained in a given a network [11, 49]. Static statistical models are common in social network analysis, as they provide a framework to compute metrics that quantify structures at different levels of organization of the nodes (individuals, groups, communities). In behavioral ecology for example, these metrics are then used to test hypotheses regarding the relationships between the attributes of a node and its position in a network (for review, see Croft et al. [18]). The most common of such hypotheses is the null hypothesis. Behavioral biologists do recognize that the use of null hypothesis significant tests requires caution [61]. In particular, it requires the definition of null hypotheses that can conceivably be true, as well as special considerations for its applications to network data. Indeed, networks represent relational data and metrics that are not independent [31], rendering statistical methods that assume independence not appropriate. To account for the dependence in relationships in network, it is believed that the best approach is to build null models based on randomizations of the networks [40, 18]. In this paper, we focus on how to generate such randomized networks that conform to an observed network.

There are many ways to randomize a given network (see [18] for a short review) whose applications are problem dependent. If the edges of the network are well established, it is possible to permute the assignments of objects with the nodes of the network, keeping the edges fixed: this enables the measure of the importance of positions within the network. In most applications however the null hypothesis focus on relationships, in which case it is preferable to randomize the edges. There are many options to perform the latter. The simplest is to assume that interactions are equally likely between nodes in the network, in which case edges can be rerranged randomly. Most real networks however are unlikely to satisfy this assumption. It is then necessary to constrain the randomization such that it conforms with the structure (geometry) of the network. This requires that the geometry of the network be known, the so-called community extraction problem [46,25,4,21], and that this geometry be translated into constraints for the random generation of edges. This task remains a challenge for the network bootstrapping community [20,43,57].

There have been many methods developed for detecting structures in complex networks [4]. Many of these methods focus on maximizing the number of relationships within communities of the network while minimizing links between communities, using algorithmic approaches [46]. Another class of methods relies on a statistical model for the network to estimate its community structure through an optimization process. The block models are probably the most common among these methods [59,51,1,63]. Most of these methods suffer from the inability to account for large fluctuations in the characteristics of the nodes that are observed in real-world complex networks (such as differences in the number of connections they make within a community, the presence of sub-geometries within communities, the presence of overlapping communities, etc.). There have been recent efforts to correct for such effects [55,28,65]. It still remains a field of study.

Traditional methods for generative models of random networks translate the geometry of the reference network in the form of constraints on the number of connections that each node make, the so-called degree of a node. This strategy has been implemented in the block models [45]; it ignores however a property observed in many real-world networks, namely clustering, or transitivity. Namely, there is a high probability that two nodes that are connected to a third node are connected to each other. There have been attempts to account for this clustering effect when generating random networks [56,15,47,37,36,38]. It remains a problem for generating random graphs.

In this paper, we propose a new approach for generating random networks that “mimic” a reference network that simultaneously resolve the challenges of detecting the structure of hierarchical community organization and embedding that geometry in the generation process. The essence of our approach lies in the meaning we give to the word mimicry: all bootstrapped networks are constructed such that they possess all (or at least most of) the global characteristics embedded within the given large network. The global characteristics of particular interest are those multi-scale features that are computable and that can ideally untangle complicated and convoluted relational patterns of cluster-wise interactions. These multi-scale patterns are best represented with a hierarchy upon the subject nodes. This hierarchy can be seen as a tree consisting of several scale levels of community decomposition. As a tree is embedded with an ultrametric, this relational hierarchy is a relational geometry. We argue from a statistical mechanics point of view that this data-driven geometry is one feasible foundation for solving the aforementioned challenges.

This paper is organized as follows. In the next section, we provide a complete overview of our bootstrapping method, underlying its foundation in capturing the ultrametric geometry of the network of interest. The following section describes the algorithms we developed to implement this method. The result section describes six applications on simulated and real undirected binary networks. We conclude the paper with a discussion on the implications of our method on network modeling, specifically on computing network entropy.

2 A general framework for geometric network bootstrapping

We focus on bootstrapping upon undirected binary networks, and leave other scenarios, such as bipartite and directed networks with binary or weighted links for separate studies.

The key principle of our method for network bootstrapping is to generate an ensemble of networks that "mimic" the original network. As such, the bootstrapped networks retain the "macroscopic features" that are captured in the so-called "macrostate" of the original network. This macrostate refers to the computed DCG-based ultrametric tree geometry on the network together with its Parisi adjacency matrix. The method includes three steps:

1. Capture the geometry of the network by building an ultrametric tree structure on its nodes,
2. Represent this tree structure using a Parisi adjacency matrix that is further optimized to minimize an Ising model-like potential field build upon the matrix,
3. Generate sampling networks that simulate each and every block of the optimized adjacency matrix.

Steps (1) and (2) are used to define the macrostate of the network, while step (3) allows for the construction of an ensemble of networks or microstates that conform to the macrostate.

Step 1: capturing the geometry of the network

A complex network that includes a large collection of nodes naturally embeds multi-scale characteristics and features, generically termed "geometry". Understanding this geometry is the foundation for extracting knowledge from the network. We have recently developed a new methodology, called data cloud geometry-tree (DCG-tree) to resolve this challenge [22,23]. We believe that this DCG-tree procedure is well suited to network analysis as (i), it automatically derives a hierarchy of clustering configurations that captures the geometric structure of the nodes, and (ii), it includes a built-in mechanism for self-correcting clustering membership across different tree levels, making it less sensitive to noise. A full description of the DCG-tree method and algorithm is provided in the original papers [22,23]. It has already been applied to studying binary networks by Chen and Fushing [16]. We note that the geometry captured by the DCG procedure is presented as a Parisi adjacency matrix [23].

We illustrate an ultrametric tree and the block structure of its corresponding Parisi adjacency matrix on a simple artificial network (Figure 1). In this example, the node-to-node distance is defined as the minimum of sums of betweenness along shortest paths, see Chen and Fushing [16] for detail. For the network illustrated in Figure 1(a), it is clear that the ultrametric tree in Figure 1(b) constitutes the multi-scale community structure, which is in turn manifested via the block structure seen in the Parisi adjacency matrix in Figure 1(c). It is noted that the interaction between two farther away core communities is represented by a corresponding block with sparsity of 1, reflecting the ultrametric distance of the tree.

Step 2: optimizing the adjacency matrix that represents the macrostate

It is not clear whether the ultrametric tree derived in step 1 actually pertains to the observed binary network data. To resolve this issue we define a potential field on the matrix lattice that is subject to a set of constraints, so called the observed degree sequence. This potential field is equivalent to a potential used in a 2D Ising model. Let $\mathscr{G}$ be the network of interest, involving N nodes. We denote as A_0 is original adjacency matrix such that $A_0(i; j) = 1$ if the nodes i and j are connected by an edge in the network and $A_0(i; j) = 0$ otherwise. The DCG algorithm applied on the network generates a Parisi adjacency matrix, A_P (step 1 above). By construction $A_P = U^*A_0U^*$ where U^* is a permutation matrix on the collection of nodes. It is known that the network is in fact invariant with respect to the permutation group, denoted as $\mathscr{U}$. Though A_0 and A_P correspond to same undirected binary network, their different arrangements of nodes on rows and columns give rise to very distinct "spin configurations" on a $n \times n$ lattice with reference to the ferromagnetic Ising model framework. Specifically, let $A(U) = [a_{ij}(U)]$ be a n n binary matrix such that $A(U) = UA_0U$ for some permutation matrix $U \varepsilon \mathscr{U}$. Since we do not allow for self-loops in a binary network, the diagonal of $A(U)$ is filled with zeros. We define an energy level of $A(U)$ as being the energy level of a ferromagnetic field having a "spin configuration" described by $A\left(U\right)+I_n=\left[a_{ij}^1\left(U\right)\right]$ with I_n the $n \times n$ identity matrix:

$$E\left[A\left(U\right)\right]=\left(-1\right)\sum_{ij}\sum_{\left(i',j'\right)\in N(i,j)}J_{<ij,i'j'>}\left(2a_{ij}^1\left(U\right)-1\right)\left(2a_{i'j'}^1\left(U\right)-1\right), \qquad (1)$$

where $N\left(i,j\right)=\left\{\left(i',j'\right)|i'=i\pm 1,j'=j\pm 1\right\}$ is the set consisting of the four nearest neighbors of $(i; j)$ entry on the lattice (mirroring extensions are required for entries on the lattice edges). The interaction potential $J_{<ij;i'j'>}$ (= 1) is taken to be constant and equal to one for simplicity. The negative constant in $E(A)$ implies that, at least heuristically, aggregations of "spins" 1′s or spins 0′s on the field of $n \times n$ lattices tend to have low energy levels, while heterogeneous spin configurations, such as checker board and its likes, give rise to high energy levels.

The (Parisi) adjacency matrix $A(U_{min})$ that achieves the globally lowest energy level under the Ising model potential field $E[A(U)]$ (equation 1) corresponds to the ground state. From a statistical mechanics perspective, this assertion implies that the corresponding ultrametric tree corresponds to a macrostate for network bootstrapping. All bootstrapped networks have to conform to such a macrostate.

We note that from an information content perspective, the optimized Parisi matrix reveals the pertinent information content of the network via its multiscale block structure. It becomes a natural platform for formulating realistic data-driven null and alternative hypotheses. Such a macrostate plays the role of an empirical distribution in classic statistical bootstrapping under the independently identically distributed (IID) setting. In this setting, the empirical distribution optimizes a nonparametric maximum likelihood in the process of estimating the unknown distribution. There has been several reports of efforts put into proving that such statistical bootstrapping works for nonparametric statistical inferences (see [19] as well as the discussion paper by DiCiccio and Efron [62]). With respect to the potential field introduced here however, we postulate a general and fundamental principle: bootstrapping conceptually has to conform to a computed data-driven macrostate of the original system, either being approximated by a network or a data cloud or others data formats.

Step 3: generating an ensemble of networks based on the macrostate representation of the network

The optimized Parisi adjacency matrix and its corresponding ultrametric tree serve as platforms for developing algorithmic computations for network bootstrapping. To ensure that the randomized sampled networks conform to the macrostate captured by the Parisi matrix, our bootstrapping algorithm works by simply sampling each block with constraints on the degrees of its nodes and patching up all the corresponding sub-networks into one large network. More specifically, the procedure follows:

i. Pick a “scale” level from the ultrametric tree. The scale defines the attention given to details in the geometry of the network. At a fine scale level, all blocks of the Parisi adjacency matrix are considered, defining communities in the network. At the large scale level, all nodes are included into a single community and the whole Parisi matrix is considered as a single block. The ultrametric tree defines the number of scales for the network of interest.

ii. Given the scale level, identify all communities and their corresponding blocks in the Parisi matrix. We consider the diagonal blocks, that include nodes within a community and the off-diagonal blocks, that define connections between communities.

iii. Generate a sample sub-network for each block, with constraints on the degrees of its nodes. The degree of a node *i* in a diagonal block *B* is set to the sum of the *ith* row element of the sub-matrix corresponding to *B* in the Parisi matrix. The degree of a node *j* in an off-diagonal block *B* between two communities *C* and *D* is set to the sum of the *jth* row or *jth* column of the corresponding sub-matrix if *j* belongs to *C* or *D*, respectively. The generating algorithms are derived from Bayati, et al. [8] and explained in details in the algorithmic section below.

iv. Regroup all sub-networks into a single network

Steps (iii) and (iv) in this procedure are repeated *M* times to generate an ensemble of *M* bootstrapped (randomized) networks. The scale level chosen in step (i) is a central concept of our procedure. By generating ensembles of networks at different scale levels along the DCG tree we can monitor how the community structure evolves. The sizes of these ensembles give us a probabilistic estimate of how likely this evolution occurs. This probabilistic interpretation has implications for network modeling and hypotheses testing, as we will discuss below. Ultimately, we want to pick the finest scale level as the corresponding bootstrapped networks will retain the geometry of the whole network (in a statistical mechanics sense, one such sampled network corresponds to a microstate that conforms to the macrostate captured by the Parisi matrix). If we pick a larger scale, the corresponding sampled network are likely to lose the structural information associated with the small communities of the original network. This is illustrated in Figure 2.

3 Algorithmic implementation

To implement the network bootstrap procedure described above we need algorithms for (i) detecting the community structure of the network of interest, (ii) optimize the Parisi adjacency matrix that represent this community structure, and (iii) build sample networks that conform to the structure captured in the optimized Parisi matrix. The following three subsections describe the algorithms we have implemented.

Detecting the community structure of the network

We solve this problem using clustering. Our algorithm proceeds in two steps. First, we define a distance between any pair of nodes in the network, following the approach described in Chen and Fushing [16], and second we cluster the nodes using this definition of distance and the DCG algorithm we recently developed [22,23]. We provide a brief description of these two steps below as they are essential for understanding our method.

Let $\mathscr{G}$ be a binary network. We write $\mathscr{G} = (\mathscr{N}, \mathscr{E})$, where $\mathscr{N} = \{n_i\}$ and $\mathscr{E} = \{e_{ij}\}$ denote its nodes and edges, respectively. The edge betweenness of an existing edge $e_{i;j}$ is denoted $b(e_{i;j})$; it is computed as being the number of shortest paths between pairs of nodes that run along $e_{i;j}$. For a pair of nodes $(n_i; n_j)$ in $\mathscr{N}^2$, we generate the set of all shortest paths in $\mathscr{G}$ $P(n_i; n_j)$ that join them. All these paths have the same length, $l_{i;j}$. We represent each of these paths as a sequence of edges, $\left\{e'_1, e'_2, ... e'_{l_{ij}}\right\}$ to which we associate the corresponding sequence of edge betweenness, $\left\{b\left(e'_1\right), b\left(e'_2\right), ..., b\left(e'_{l_{ij}}\right)\right\}$. The distance between the two nodes n_i and n_j is then defined as:

$$d\left(n_i n_j\right) = \underset{\mathscr{P}(n_i,n_j)}{min} \sum_{k=1}^{l_{ij}} H\left(b\left(e'_k\right)\right) \quad (2)$$

where $H(\cdot)$ is a monotonically increasing kernel function. In Chen and Fushing (2012), we established that this distance is usually small for a pair of nodes that belong to the same community, and large for a pair of nodes that belong to different communities. It is therefore appropriate for community detection in networks using clustering.

Starting from the set of nodes $\mathscr{N}$ of the network $\mathscr{G}$ and the complete distance matrix $\mathscr{D}$ over these nodes computed with equation 2, we generate a hierarchical tree structure using the DCG clustering technique. The main idea of the DCG method is to embed the geometry defined by $\mathscr{D}$ into a ferromagnetic potential landscape; its implementation is then based on two key observations. First, it is observed that $\mathscr{D}$ defines a weighted graph onto $\mathscr{N}$. By equating the weight on an edge with a ferromagnetic potential, this weighted graph is seen as equivalent to a potential landscape, typically characterized by many wells with various depths. Second, it is possible to explore this landscape and therefore define its geometry by using the popular dynamic Monte Carlo approach. A random walk as a function of “time” will identify the many wells of the potential, as well as the probability of jumping from one well to another. An additional advantage of using dynamic Monte Carlo is that it provides a different dimension to explore the geometry of the landscape, characterized with its temperature parameter *T*. To benefit from the latter, we define the ferromagnetic field such that it places the potential $w_{ij} = e^{-d_{ij}/T}$ on link e_{ij} between nodes n_i and n_j on the graph, where *T* is a parameter mimicking temperature. At a high temperature *T*, a Markovian walk on the energy landscape will transition from any node to most of the other nodes with more or less equal probabilities. At a low temperature however, the Markov chain tends to get trapped in potential wells for various periods of time depending on the sizes of the well before it can escape. These two observations led to the following two-device algorithm, named Data Cloud Geometry or DCG, for deriving the underlying multi-scale geometry of a network. At a given temperature *T*, a regulated random walk on the equivalent ferromagnetic landscape as a function of time detects information about the number of clusters and the corresponding cluster membership of individual nodes. By repeating this procedure at different temperatures, the DCG algorithm derives the geometric hierarchy of the set of nodes $\mathscr{N}$, as described in Fushing and McAssey [22]. These temperatures are then taken as energy barrier heights to define an ultrametric topology onto the network as it is a system on a ground state. This topology provides measurable and natural distances between clusters. The ultrametric topological space can then be summarized as a hierarchical tree, the DCG-tree [23]. This ultrametric tree can be represented as a Parisi adjacency matrix by simply re-arranging its rows and columns so that it is consistent with the tree structure.

Optimizing the adjacency matrix

The Parisi adjacency matrix A_P obtained from step 1 is further optimized by first defining a potential field *E* on the matrix according to equation 1 and then by searching for the permutation *U* that leads to a minimum of *E*:

$$U^* \approx arg\,\min_{U \in \mathscr{U}} E\left[A\left(U\right)\right], \quad (3)$$

We propose algorithm 1 as a pragmatic resolution to find *U**.

It is clear that the DCG-tree provides the initial arrangement of nodes for the initial version of Parisi adjacency matrix into blocks (step (1)). Step (2) is designed to arrange the corresponding blocks into order to achieve lower energy level, while Step (3) is set to arrange the nodes within blocks such that an even lower energy level may be reached. All arrangements are done subject to the DCG-tree structure. We note that a DCG-tree is invariant with respect to switching arrangements between branches on the same levels, and to switching arrangements among nodes within the lowest branches. Step (4) is an attempt to evaluate the possibility of being too far away from the ground state. From a computational perspective, the critical idea behind this algorithm is that the DCG-tree, or its DCG-ultrametric tree, provides a “natural” distance metric in the space of permutation $\mathscr{U}$.

We take the final permutation, $\hat{U}^*$, to be an approximation of U^*. If step (4) in the algorithm does not lead to reduction of the energy, or if this reduction is minimal, we set $U = U_2$. If step (4) however achieves significant reduction of the energy level, we need to reconstruct a new ultrametric tree and repeat the whole procedure. This was not necessary in all the test cases presented below. This is by itself not surprising as a low energy Parisi adjacency matrix is located in a potential well near the ground state. Computer experiments have confirmed that the process of perturbation implemented in Step (4) tends to push the Parisi matrix out of the potential well and then wander over the rough and higher energy landscape. This phenomenon indicates that the betweenness based distance metric used in the DCG-based ultrametric geometry is reasonable.

Generating random networks that conform to the optimized macrostate

The optimized Parisi adjacency matrix and the corresponding DCG tree serve as support for generating bootstrapped networks. The generation itself is performed according to step 3 described in the previous section. Briefly, we pick first a scale level from the DCG tree (usually the finest level if we want to generate networks that conform to the structure of the original network), we isolate the corresponding diagonal and off diagonal blocks from the Parisi matrix, we set the degrees of the nodes belonging to these blocks based on the sums of their corresponding rows or columns in the block, we generate random sub-networks for all blocks that are constrained by these degree sequences, and finally we combine these sub-networks into the final bootstrapped network. While most of these steps are straightforward, we need an algorithm for generating a random network on a set of nodes that conforms to constraints on the degrees of these nodes. We use for this purpose an algorithm proposed by Bayati, et al. [8], which is derived from works from McKay [41], Steger and Wormald [60], and Kim and Vu [29]. We give our version of this algorithm for a generic diagonal block of the Parisi matrix involving the set of nodes *A* as algorithm 2.

Algorithm 2 is applied to all diagonal blocks of the Parisi matrix, and a slightly modified version is used to generate the bi-partite graphs corresponding to the off-diagonal rectangle blocks. The number *N* returned by the algorithm is an estimate of the size of the ensemble of corresponding random graphs that can be generated for the block considered. The size of the ensemble of bootstrapped networks for the whole Parisi matrix is then computed by multiplying the estimated sizes of the sub-networks generated from all blocks considered.

In algorithm 2, the probability of picking a pair of nodes $(a_i; a_j)$ is taken to be:

$$p_{ij} \propto \left(\hat{d}a_i \hat{d}a_j\right)^{\alpha} \left(1 - \frac{da_i da_j}{4m}\right) \quad (4)$$

In the original version of this algorithm proposed by Bayati, et al. (2010) the exponent α is taken to be equal to 1. We noticed however that when one or multiple hubs are present in the block considered, the algorithm often fails to generate a network that fully satisfies the constraints on the degrees of the nodes. To resolve this problem, we defined:

$$\alpha = 1_{[|d_i - d_j| > k]} \quad (5)$$

that is, α an indicator function with an empirically chosen thresholding exponent *k* (i.e. α = 1 if $\{d_i - d_j| > k$ and 0 otherwise). This is a pragmatic correction that circumvents the failure problem, with the side effect of underestimating the size of the corresponding ensemble.

We note that there are other algorithms described in the literature for generating rectangle binary matrices with prescribed row and column sums (see for example [17] and [5]). These algorithms rely on the maximum entropy property and provide asymptotic estimates of the size of the ensemble of random networks that can be generated. The algorithm proposed here has the advantage to be simple to implement.

4 Simulation studies

We report the application of our method to generate ensemble of randomized networks on one simulated, and four real undirected binary networks. We note that in the latter cases, the actual geometry of the network is not known; our discussion is correspondingly more qualitative than quantitative.

4.1 Simulated network: a symmetric case with 21 motifs

This simulated network, used in Chen and Hsieh [16], consists of 21 cliques each with four nodes, with a symmetric geometry illustrated in Figure 3. The DCG-tree analysis of this network reveals a geometry with 3 scales: a fine scale that identifies the 21 cliques (Figure 3(a)), a medium scale with 4 communities, corresponding to the four branches of the structure built in the network, with the middle clique assigned to one of the branches (Figure 3(b)), and a large scale with all 21 cliques regrouped into a single community (Figure 3(c)). When we generate networks based on the fine scale, we find a single solution, the network itself (Figure 3(d)); this indicates that the network defined as a union of 21 communities is deterministic. When we generate networks at the medium scale corresponding to 4 communities, we start seeing connections between the cliques that belong to the same branch, i.e. we lose the internal geometry of these branches in the bootstrapped network (see one example of such bootstrapped network in Figure 3(e)). The effect is even more prominent as we move to the large scale with a single community: the sampled networks start losing the branched geometry of the network (see an example of such bootstrapped network in Figure 3(f)).

The qualitative differences observed between the networks generated at the three different scales mentioned above can in fact be quantified with an entropy measure. A key feature of algorithm 2 that we use to generate the sampling networks is that it also provides an estimate of the size of the ensemble of networks that satisfy the degree constraints (see [8] for a full discussion of this property). We can use this estimate of size to compute an estimate of the entropy of the ensemble (where the entropy is simply the logarithm of the size). This allows us to compute the entropy evolution of the network as it evolves from a fine structure with 21 communities to coarse structures with 4 community and ultimately 1 community. The corresponding values are 0.3, 261.7, and 474.3, respectively. The near zero entropy for the 21 community representation of the network confirms that this geometry is nearly deterministic. The large jumps in entropy as the network evolves to coarser geometry with less communities indicates that at these scales, the network allows for larger geometric distortions, as illustrated in Figure 3.

4.2 Real networks

The karate club network—A karate club was observed for a period of three years, from 1970 to 1972 [64]. A formal model of the friendship network within this club, one that contained sufficient complexity and precision to allow the testing of propositions was constructed.

This network is often considered as a benchmark for evaluating the effectiveness of community detection algorithms. The analysis of this network with DCG reveals three stable scales, a fine scale with three communities, a medium scale with two communities, and a coarse, or large scale with all nodes regrouped in a single community (see Figures 4(a), 4(b), and 4(c), respectively There is one additional outlier (i.e. a community with a single node) that is consistent in the fine and medium scales; this node is shown as white in Figures 4(a) and 4(b). We generated ensembles of random networks that conform to these three scales; examples of these ensembles are shown in Figures 4(d), (e), and (f). It is clear that these networks conform to the corresponding community geometry. The entropies of the ensembles nearly double as we move from one scale to the other, with values of 29.5 for the fine level, 64.9 for the medium level, and 122.4 for the coarse level. This is equivalent to saying the complexity of this network doubles as we merge two of its communities.

The merging patterns observed in the Karate club example differ from the communities observed in other analyses of the same data set; we believe however that they are sound. Girvan and Newman [24] for example reported only two communities for the Karate club, one that maps with the community illustrated in blue and a second one that maps with the combined red and green communitites (see Figure 4(a)). It is reasonable to consider however that the smallest “green” community stands alone as its members are all newcomers who established links to the club through the “instructor”, who appears as a hub in the “red” community. The more senior members of the Karate club belong to the two larger communities and they share many inter-community links. In contrast, there are no inter-community links between members of these large communities with members of the small “green” communities, the newcomers. This observation is captured by our procedure, as the two large communities merge first in the ultrametric tree we build on the Karate Club network.

An 8-letter doublets words network—There is a popular English word game called “Doublets”, which was created by the English author Charles Lutwidge Dodgson, the author of “Alice’s Adventures in Wonderland (1865)” under the pseudonym Lewis Carroll. A doublets network can be constructed based on this game. All English words serve as nodes of this network and an edge is drawn between two words if they share the same alphabetic letters but one. For instance, the words DIVE, DIRE, WIRE, and WIPE form a chain of edges. Connected components of the whole doublets network contain words of the same length. Here we consider the largest clique-component of 8-letter words; it contains 291 nodes.

Analysis of this network with DCG reveals three stable scales: a fine scale with four communities (Figure 5(a)), a median scale level with two communities (Figure 5(b)), and a coarse level including all nodes (Figure 5(c)). Examples of random networks that conform to this network at these three levels of community structures are shown in Figures 5(d), 5(e), and 5(f). Interestingly, the corresponding ensembles do not vary significantly in size: their entropies are 1291, 1521, and 1869 for the fine, medium, and coarse level, respectively. This would indicate that there is only little specificity in the geometry of the different communities detected in the network.

Movie-star network—There is large movie and actor data set available at the URL: http//www3.nd.edu/networks/resources.htm... This data set contains 102,639 movies and 376,021 actors in the form raw data. In this study we consider the connectivity among “movie-stars” only. A movie-star is an actor that appears at least in 100 movies. There are 747 movie-stars in this data set. Two movie-stars are considered to be connected (i.e. to form an edge in the movie-start network) if they have ever performed together in at least one movie. The corresponding binary network is a network of hubs. We use this example to illustrate the concept of “scale” and “heterogeneity” in a large network data. That is, we establish the hub-vs-hub relationship first. Then its information patterns can be contrastingly compared with hub-vs-non-hub relationship patterns via a bipartite network.

The multiscale structure of the network is clearly visible from the different heat maps shown in Figure 6. The original adjacency matrix (noted as A_0 in Section 2) has an energy $E[A_0] = -981516$. After imposing the DCG-tree geometry onto A_0, a Parisi adjacency matrix (noted as A_P in algorithm 1) is obtained; its energy is −1267084. After permutation of the blocks in A_P, we obtain a new matrix (noted $A_P(U_1)$), whose energy is −1289456. The reduction of energy for the different Parisi adjacency matrices generated with algorithm 1 are consistent with the assertion that we have reached the macrostate of the binary network.

World-wide major airport network—Open Flights is a tool that allows its users to map flights around the world, and to search and filter them, thereby providing valuable information about air traffic. It is available at the URL http//www.openflights.org. Using this tool, we generated information about airports and flights for the year 2012 from the original OpenFlights/Airline Route Mapper Route Database, which contains 59036 routes between 3209 airports on 531 airlines spanning the globe. We then defined “major airports” as airports from which at least ten direct flights originate. There are 668 such major airports. Two major airports are connected if and only if there is a direct flight connects them. This definition of connectivity leads to a network with a total number of edges 10861 and a mean degree for its vertices of 32.5. We use this example is to illustrate the concordance between known landmark information for the vertices of a network and the DCG-based ultrametric geometry pertaining to the network. In Figure 7, we show two levels of the ultrametric tree computed on the major airport network using DCG, as two different temperature. At a relatively low temperature (Figure 7a), twelve clusters are identified; these clusters match reasonably well with geographic landmarks.

Major airports in America continent form three clusters that cover three distinct areas that are geographically consistent: 1) Alaska Area; 2)North and Central America; 3) and south

America, for the lower temperature level of the DCG-based ultrametric geometry. Only the first two clusters merge at higher temperatures, while the cluster covering South America remains constant. Major airports in Europe and Africa form two clusters matching with a Europe-Africa Region and a Russia-Eastern Europe Region. These two clusters merge with the airport cluster covering East Asia at a higher temperature. Across the two temperatures, the cluster of major airports covering Southern Asian-Australian Region remains separated from the other clusters. These matching patterns between clusters and geographic landmarks together with their emerging patterns support the pertinence of our DCG-based ultrametric geometry.

5 Discussion

Statistical physics has been very successful in deducing interplays of fundamental laws to explain complex phenomenons in nature. In this manuscript, we illustrate its application on complex networks, which are nowadays popularly used for approximating complex systems.

Our first key development is that a network is viewed and properly redefined as a thermodynamic system equipped with an Ising potential energy. Its ground state is then postulated to be the state that contains the actual network pattern information. The task of computing this information is thereby converted into the task of finding its ground state, i.e. a discrete combinatorial optimization problem. It is well known that the computational complexity of solving this problem via a direct search algorithm is untraceable, especially when the number of nodes in the network is large. This leads to our second key development: to resolve this computational complexity by constructing a data-driven geometry on the node space such that nearly all non-viable up-and-down spin configurations are automatically excluded from our search algorithm. This device is an ultrametric (tree) geometry based on Data Cloud geometry (DCG) computations [22].

The data-driven tree geometry that is constructed with the DCG method reveals patterns of multiscale blocks on the adjacency matrix representation of the network. This multiscale block pattern is consistent with the Parisi ultrametricity conjecture [52,53]. We consequently refer to this matrix representation as the Parisi adjacency matrix; it is assumed to correspond to the macrostate of the network. The concept of multiscale features in a network is consistent with the hierarchical structure identified by Herbert Simon [26] as well as with the concept of scales and complexity suggested by Philip. W. Anderson [3]. Henceforth we are confident that an authentic network structure is necessary multiscale. This has been emphasized in [38,16,23]. Any microstate of the thermodynamic system pertaining to the network of interest has therefore to conform to this macrostate. Four our purpose, this translates to the fact that any random network designed to mimic a given network has to conform with the multiscale block structure of this network.

We emphasize the need to build an ultrametric geometry for a network. This geometry is data-driven and nonparametric. as such, it is not built from any modeling or embedding assumption. The DCG-based construction starts from a “betweenness” based distance, in which basic network characteristics, such as node degrees and centrality, are contained. Nodes sharing same motifs, or core clusters, tend to have similar betweenness. The whole network is then somehow properly summarized into a distance matrix. Furthermore the different levels in the resultant DCG-tree structure are correspondingly derived to embrace major phase-transitions of the cluster-merging process.

We note that this data-driven geometry contrasts sharply with the concept of hyperbolic geometry embedded onto a network [12,13,35,34,14,33]. The hyperbolic geometry is built by theoretically distributing nodes randomly on a R^2 disk, for instance, with a tuning constant curvature and is taken as a “continuous and smooth” version of a “tree” geometry. Though this hyperbolic geometry of network also bring out many characteristics, such as degree heterogeneity, strong clustering and many power-laws related manifestations, its structural features and patterns are difficult to visualize.

To search for the ground state of the network, we optimize its adjacency matrix A_P such that it leads to a minimum for an Ising-like 2D potential E on the network (equation 1. To perform this optimization, we consider the group of permutations U on A_P and finds the permutation of the rows and columns that lead to a minimum of E (equation 3). We note that all permutations in the permutation group, including the optimal permutation U, keep the original binary network unaltered. If perturbations on the original network were allowed, for instance subject to degree sequences pertaining to any level of the Parisi adjacency matrix, then it is expected that we could find even lower energy levels. In general however, it is difficult to search directly the ensemble of matrices subject to a set of observed degree sequences as this ensemble is far too large. The algorithm we proposed (algorithm 1) provides an effective computational solution to this problem. From a statistical mechanics perspective, it is reasonable to conjecture that all the spin-glass states of the minimum energy Hamiltonian are represented by a collection of Parisi adjacency matrices. To find the optimal matrix, we need only to consolidate the connectivity of on-diagonal blocks pertaining to the finest scale level. One approach to obtain this consolidation is the so called checkerboard unit switch found in ecology literature (see for example [39]). Such a functional switch would reduce the energy level.

Once the geometry of a network is known, we use an algorithm originally proposed by Bayati and colleagues [8] to generate random networks that conform to this geometry. There are two primary reasons for using this algorithm: (i) its capability of generating nearly “uniform” random matrices with computational simplicity, and (ii) its feature of providing an estimate for the size of the ensemble of random networks that can be generated. A simple algorithm that can achieve uniformity is typically ideal because of its computational reliability. This is an essential aspect for implementing network bootstrapping in statistical inferences. We note that the popular standard configuration model in random graph literature is not a viable alternative for the algorithm we have used, as it is only capable of generating undirected random graph with a fixed degree distribution [42,50,38].

The generation of random networks that conform to the geometric structure of the network can be repeated at the different levels encoded in this geometry. As the algorithm we use provides an estimate of the sizes of these different ensembles of networks, we can compute the evolution of the entropy of the network with respect to its multiscale block structure. From a practical perspective, this entropy evolution provides a mean to test whether a scalespecific block structure is realistic or not. This is one of the main applications of network bootstrapping in network based hypothesis testing as commonly encountered in biology and ecology (for review, see Croft, et al. [18]). We believe that the method we proposed here is well adapted to such testing. The Parisi adjacency matrix and its several levels of network ensembles would not only provide a platform for formulating null and alternative hypotheses, but also pave the way for proper nonparametric statistical inferences.

Bianconi [9,10] recently proposed an alternate method for computing the complexity of randomized ensembles of networks that is also based on statistical mechanics. In her reports, she considered three types of such randomized ensembles with either fixed degree sequence, degree-degree correlation, or community constraints. She then proposed a method for approximating the partition function of these networks, using standard tools from mathematical physics. An entropy is computed based on the partition function and this entropy is used to define an index of network complexity under the corresponding structural constraint. In contrast, the method we have proposed is data-driven and as such, it alleviates the problem of estimating a partition function. We rely on the structural patterns of the network to generate ensembles of randomized networks that capture the different scales present in the network; the sizes of these ensemble provide not only a measure of the complexity of the network itself, it provides a measure of its evolution as we move from its coarse to fine community content.

Acknowledgments

This work was partially supported by National Science Foundation Grant DMS-1007219 (co-funded by Cyber-enabled Discovery and Innovation (CDI) program). PK acknowledges support from the NIH.

References

1. Airoldi EM, Blei DM, Fienberg SE, Xing EP. Mixed membership stochastic blockmodels. J. of Machine Learning Research. 2008; 9:1981–2014.
2. Albert R, Barabási AL. Statistical mechanics of complex networks. Rev. Mod. Phys. 2002; 74:47–97.
3. Anderson PW. More is different. Science. 1972; 177:393–396. [PubMed: 17796623]
4. Barabási AL. Scale-free networks: a decade and beyond. Science. 2009; 325:412–413. [PubMed: 19628854]
5. Barvinok A. On the number of matrices and a random matrix with prescribed row and column sums and 01 entries. Adv. Math. 2010; 224:316–339.
6. Bascompte J. Disentangling the web of life. Science. 2009; 325:416–419. [PubMed: 19628856]
7. Bascompte J. Structure and dynamics of ecological networks (perspective). Science. 2010; 329:765–766. [PubMed: 20705836]
8. Bayati M, Kim JH, Saberi A. A sequential algorithm for generating random graphs. Algorithmica. 2010; 58:860–910.
9. Bianconi G. The entropy of randomized network ensembles. Europhys. Lett. 2008; 81:28005.
10. Bianconi G. Entropy of network ensembles. Phys. Rev. E. 2009; 79:036114.
11. Boccaletti S, Latora V, Moreno Y, Chavez M, Hwang DU. Complex networks: Structure and dynamics. Phys. Rep. 2006; 424:175–308.
12. Boguñá M, Krioukov D. Navigating ultrasmall worlds in ultrashort time. Phys. Rev. Lett. 2009; 102:058701. [PubMed: 19257567]
13. Boguñá M, Krioukov D, Claffy KC. Navigability of complex networks. Nat. Phys. 2009; 5:74–80.

14. Boguñá M, Papadopoulos F, Krioukov D. Sustaining the internet with hyperbolic mapping. Nat. Commun. 2010; 1:62. [PubMed: 20842196]

15. Bollobás B, Janson S, Riordan O. Sparse random graphs with clustering. Random. Struct. Algor. 2011; 38:269–323.

16. Chen C, Fushing H. Multi-scale community geometry in network and its application. Phys. Rev. E. 2012; 86:041120.

17. Chen Y, Diaconis P, Holmes SP, Liu JS. Sequential monte carlo methods for statistical analysis of tables. J. Am. Stat. Assoc. 2005; 100:109–120.

18. Croft D, Madden J, D.W. F, R. J. Hypothesis testing in animal social networks. Trends Ecol. Evol. 2011; 26:502–507. [PubMed: 21715042]

19. Efron B. Bootstrap methods: Another look at the jackknife. Ann. Statist. 1979; 7:1–26.

20. Fallani FDV, Nicosia V, Latora V, Chavez M. Nonparametric resampling of random walks for spectral network clustering. Phys. Rev. E. 2014; 89:012802.

21. Fortunato S. Community detection in graphs. Phys. Rep. 2010; 486:75–174.

22. Fushing H, McAssey M. Time, temperature and data cloud geometry. Phys. Rev. E. 2010; 82:061110.

23. Fushing H, Wang H, Van der Waal K, McCowan B, Koehl P. Multi-scale clustering by building a robust and self-correcting ultrametric topology on data points. PLoS ONE. 2013; 8:e56259. [PubMed: 23424653]

24. Girvan M, Newman MEJ. Community structure in social and biological networks. Proc. Natl. Acad. Sci. (USA). 2002; 99:7821–7826. [PubMed: 12060727]

25. Goldenberg A, Zheng ZX, Fienberg SE, Airoldi EM. A survey of statistical network model. Foundations and Trends in Machine Learning. 2009; 2:1–117.

26. H. S. The architecture of complexity. Proc. Am. Phil. Soc. 1962; 106:467–482.

27. Havlin, S.; Cohen, R. Complex networks: structure, robustness, and function. Cambridge University Press; Cambridge, UK: 2010.

28. Karrer B, Newman M. Stochastic blockmodels and community structure in network. Phys. Rev. E. 2011; 83:016107.

29. Kim, J.; Vu, V. Proc. ACM Symp. on Theory of Comput. (STOC). 2003. Generating random regular graphs; p. 213-222.

30. Kolacyzk, E. Statistical analysis of network models. Springer; New York: 2009.

31. Krakhardt D. Predicting with networks: non parametric multiple regression analysis of dyadic data. Soc. Networks. 1988; 10:359–381.

32. Krause J, Croft R, James R. Social network theory in the behavioural sciences: potential applications. Behav. Ecol. Sociobiol. 2007; 62:15–27.

33. Krioukov D, Kitsak M, Sinkovits R, Rideout D, Meyer D, Boguñá M. Network cosmology. Sci. Rep. 2012; 2:793. [PubMed: 23162688]

34. Krioukov D, Papadopoulos F, Kitsak M, Vahdat A, Boguñá M. Hyperbolic geometry of complex networks. Phys. Rev. E. 2010; 82:036106.

35. Krioukov D, Papadopoulos F, Vahdat A, Boguñá M. Curvature and temperature of complex networks. Phys. Rev. E. 2009; 80:035101.

36. Lancichinetti A, Fortunato S. Community detection algorithm: A comparative analysis. Phys. Rev. E. 2009; 80:056117.

37. Lancichinetti A, Fortunato S, Radicchi F. Benchmark graphs for testing community detection algorithms. Phys. Rev. E. 2008; 78:046110.

38. Lancichinetti A, Radicchi F, Ramasco JJ, Fortunato S. Finding statistically significant communities in networks. PLoS One. 2011; 6:e18961. [PubMed: 21559480]

39. Manly B. A note on the analysis of species co-occurrences. Ecology. 1995; 76:1109–1115.

40. Manly, B. Randomization, Bootstrap, and Monte Carlo methods in biology. CRC Press; , Boca Raton, Fl: 2006.

41. McKay B. Asymptotics for symmetric 0-1 matrices with prescribed row sums. Ars. Combinatoria A. 1985; 19:15–25.

42. Molloy M, Reed B. A critical point for random graphs with a given degree sequence. Random Strut. Algor. 1995; 6:161–179.

43. Musmeci N, Battiston S, Caldarelli G, Puliga M, Gabrielli A. Bootstrapping topological properties and systemic risk of complex networks using the tness model. J. Stat. Phys. 2013; 151:720–734.

44. Newman M. The structure and function of complex networks. SIAM Reviews. 2003; 45:167–246.

45. Newman M. Detecting community structure in networks. Eur. Phys. J. B. 2004; 38:321–330.

46. Newman M. Modularity and community structure in networks. Proc. Natl. Acad. Sci. (USA). 2006; 103:8577–8582. [PubMed: 16723398]

47. Newman M. Random graphs with clustering. Phys. Rev. Lett. 2009; 103:058701. [PubMed: 19792540]

48. Newman, M. Networks: an introduction. Oxford University Press; Oxford: 2010.

49. Newman, MEJ.; Barabási, AL.; Watts, DJ. The Structure and Dynamics of Networks. Princeton Univ. Press; New Jersey: 2006.

50. Newman MEJ, Girvan M. Finding and evaluating community structure in networks. Phys. Rev. E. 2004; 69:026113.

51. Nowicki K, Snijders TAB. Estimation and prediction for stochastic blockstructures. J. Am. Stat. Assoc. 2001; 96:1077–1087.

52. Parisi G. In nite number of order parameters for spin-glasses. Phys. Rev. Lett. 1979; 43:1754–1756.

53. Parisi G. A sequence of approximate solutions to the S-K model for spin glasses. J. Phys. A. 1980; 13:L115–L121.

54. Proulx S, Promislow D, Phillips P. Network thinking in ecology and evolution. Trends Ecol. Evol. 2005; 20:345–353. [PubMed: 16701391]

55. Reichardt J, Alamino R, Saad D. The interplay between microscopic and mesoscopic structures in complex networks. PLoS One. 2011; 6:e21282. [PubMed: 21829597]

56. Rosvall M, Bergstrom CT. Maps of random walks on complex networks reveal community structure. Proc. Natl. Acad. Sci. (USA). 2007; 105:1118–1123. [PubMed: 18216267]

57. Rosvall M, Bergstrom CT. Mapping change in large networks. PLoS One. 2010; 5:e8694. [PubMed: 20111700]

58. Sih A, Hauser S, McHugh K. Social network theory: new insights and issues for behavorial ecologists. Behav. Ecol. Sociobiol. 2009; 63:975–988.

59. Snijders TAB, Nowicki K. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. J. Classi cation. 1997; 14:75–100.

60. Steger A, Wormald NC. Generating random regular graphs quickly. Comb. Prob. Comput. 1999; 8:377–396.

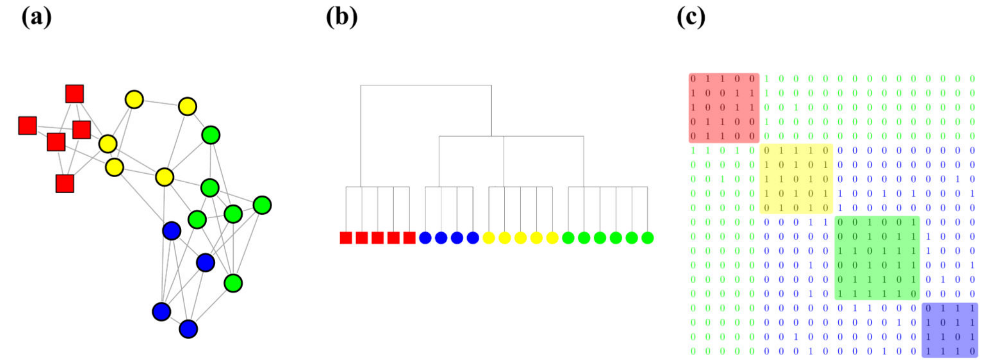
61. Stephens P, Buskirk S, del Rio C. Inference in ecology and evolution. Trends Ecol. Evol. 2007; 22:192–197. [PubMed: 17174005]

62. T.J D, B E. Bootstrap con dence intervals (with discussion). Statistical Science. 1996; 11:189–228.

63. Wainwright MJ, Jordan MI. Graphical models, exponential families and variational inference. Machine Learning. 2008; 1:1–305.

64. Zachary WW. An information ow model for con ict and ssion in small groups. J. Anthropol. Res. 1977; 33:452–473.

65. Zhao Y, Levina E, Zhu J. Community extraction for social networks. Proc. Natl. Acad. Sci. (USA). 2011; 108:7321–7326. [PubMed: 21502538]



$$\begin{matrix}
0&1&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&1&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&0&1&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&1&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&1&1&0&1&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&1&0&1&0&1&1&0&0&1&0&1&0&0&0&1\\
0&0&0&0&0&0&1&0&1&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&0&0&1&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&1&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&1&0&1&1&1&1&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&1&0&1&1&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&1&0&1&0&0\\
0&0&0&0&0&0&0&0&1&0&1&1&1&1&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&1\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&1&0&1\\
0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&1&1&1&0
\end{matrix}$$

Fig. 1. A network and its hierarchical structure

(a) A undirected binary network, (b) the DCG-based ultrametric tree built on its nodes, and (c) the corresponding Parisi adjacency matrix. Color online.

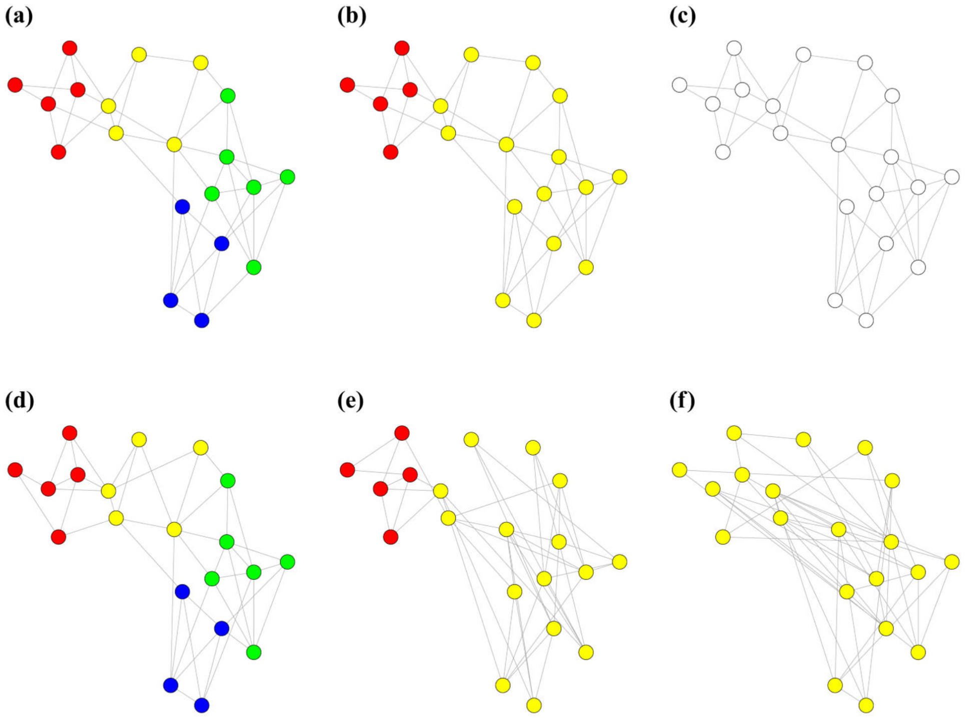


Fig. 2.

Effects of the scale level on the quality of the bootstrapped networks. Let us consider a simple binary network with 20 nodes. The DCG procedure applied to its nodes identifies three scales, fine (a), medium (b), and large (c), with 4, 2, and 1 communities, respectively. A random network selected to conform to the fine scale level maintains the structure of the network (panel d). Random networks that conform to the medium scale and to the large scale shown in panels (e) and (f), respectively, lose the geometric information from the small communities that are not identified at those scale levels, such as the “blue” community. Color online.

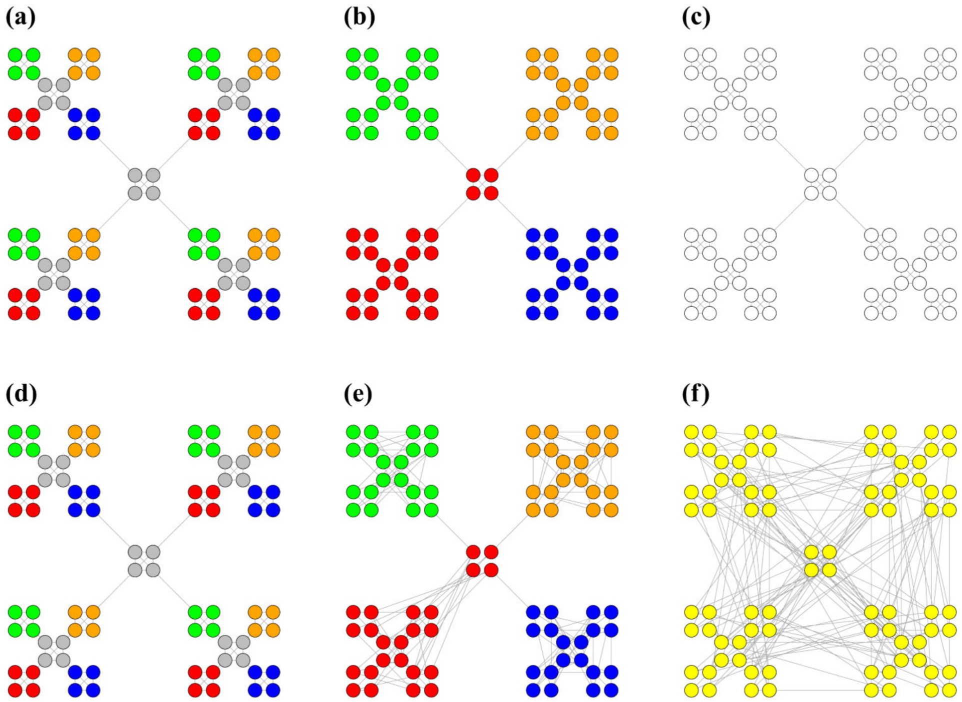


Fig. 3. The 21-motif example

This network was designed to illustrate the importance of geometry when sampling networks. The DCG analysis of this network reveals three stable scale levels, with 21 (a), 4 (b), or 1 (c) communities, respectively. In each of the panels (d), (e), and (f), we show one example of a bootstrapped network at the corresponding scale levels. While the network is nearly deterministic at the ne level (d), it becomes blurry at the large level (f). See text for details. Color online.

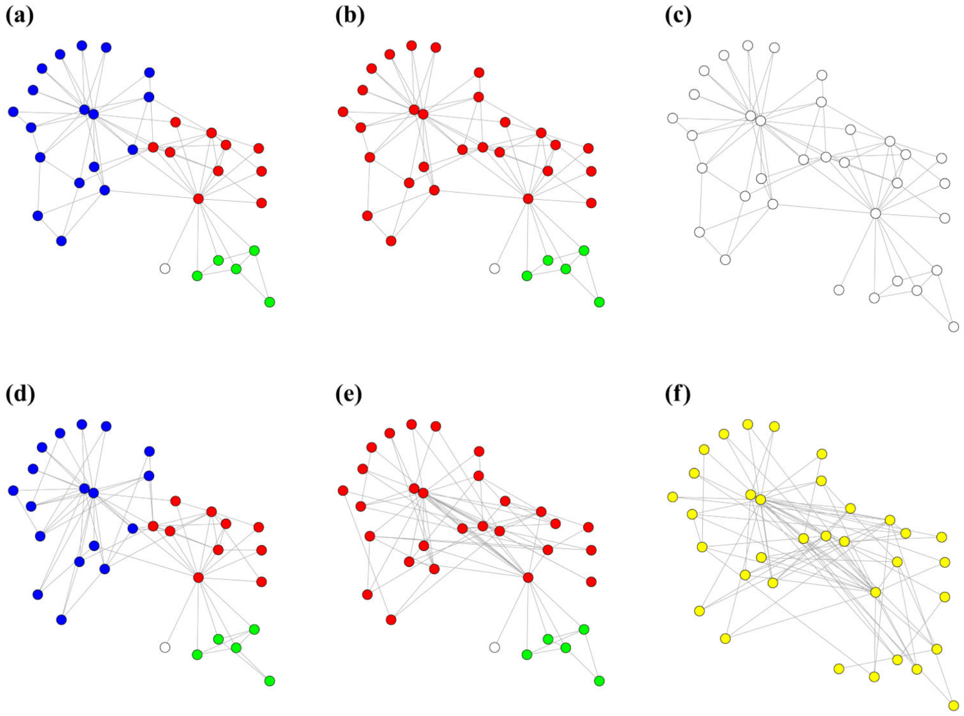


Fig. 4. The Karate Club friendship network (Zachary, 1977)

DCG analysis of this network reveals a geometry with three levels, one with three communities (a), one with two communities (b), and one where all nodes belong to the same community (c). In panels (d), (e), and (f), we show examples of randomized networks that were generated to conform to the karate club network at these levels. Color online.

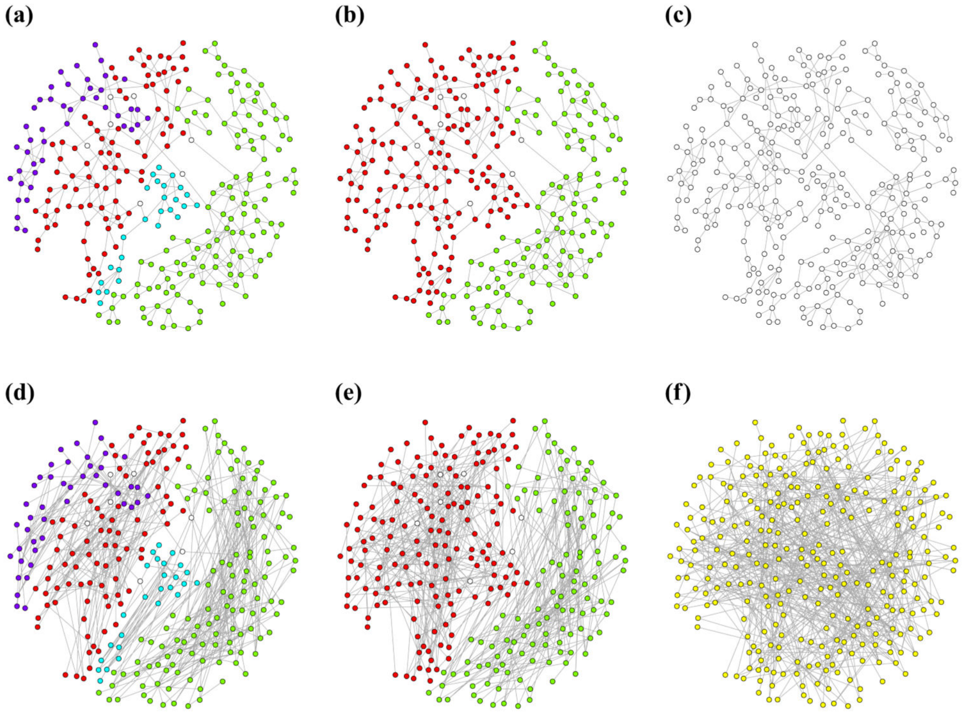


Fig. 5. A 8-letter word doublets network

Panels (a), (b), and (c) show the network with 4, 2, and a single communities, respectively, where the communities were identified using DCG. In each of the panels (d), (e), and (f), we show one example of a bootstrapped network at the corresponding scale levels. Color online.

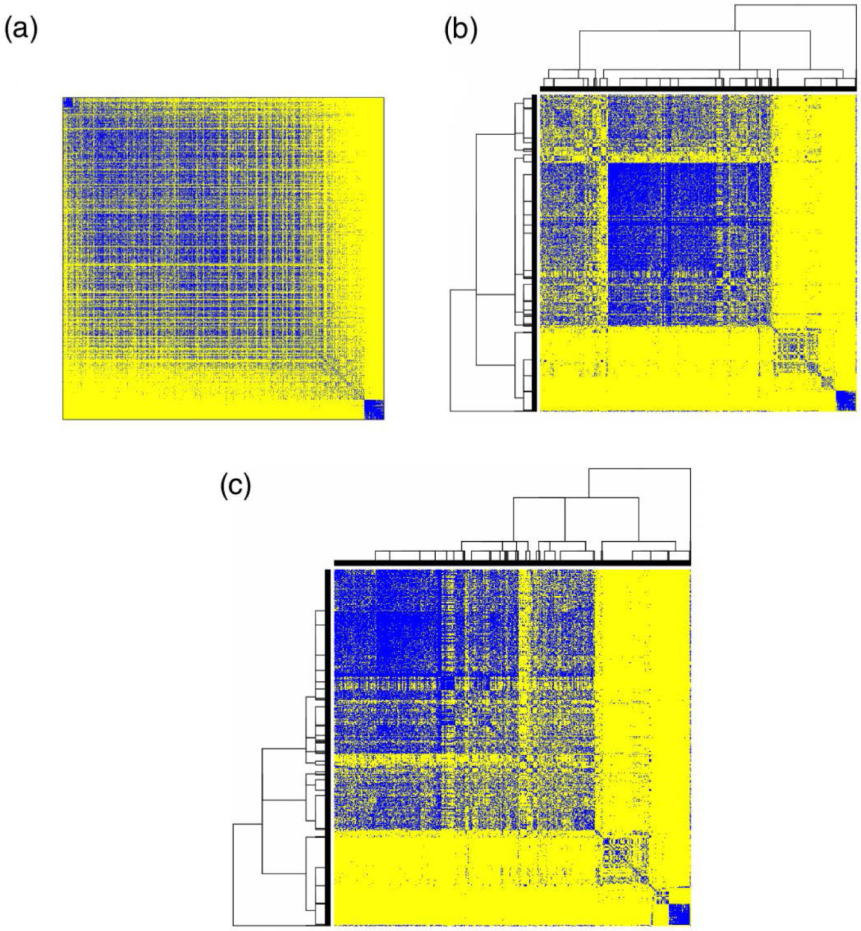


Fig. 6.

Parisi adjacency matrices for the network of 747 Movie stars, who were actors in more than 100 movies (see text for details). (a) The heatmap of the network original adjacency matrix A_0; (b) The heatmap of the Parisi adjacency matrix A_P; (c) The heatmap of the Parisi matrix A_P (U_1) obtained after permutations of the blocks of A_P; this matrix has a lower energy, namely −1289456. Color online.

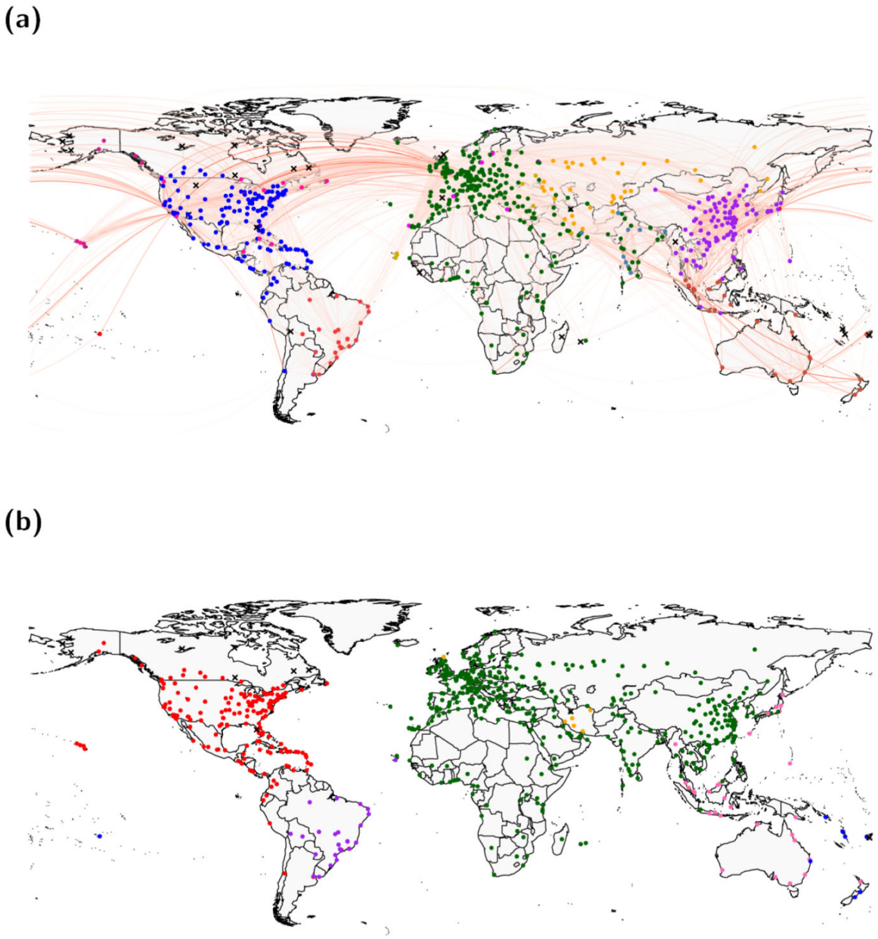
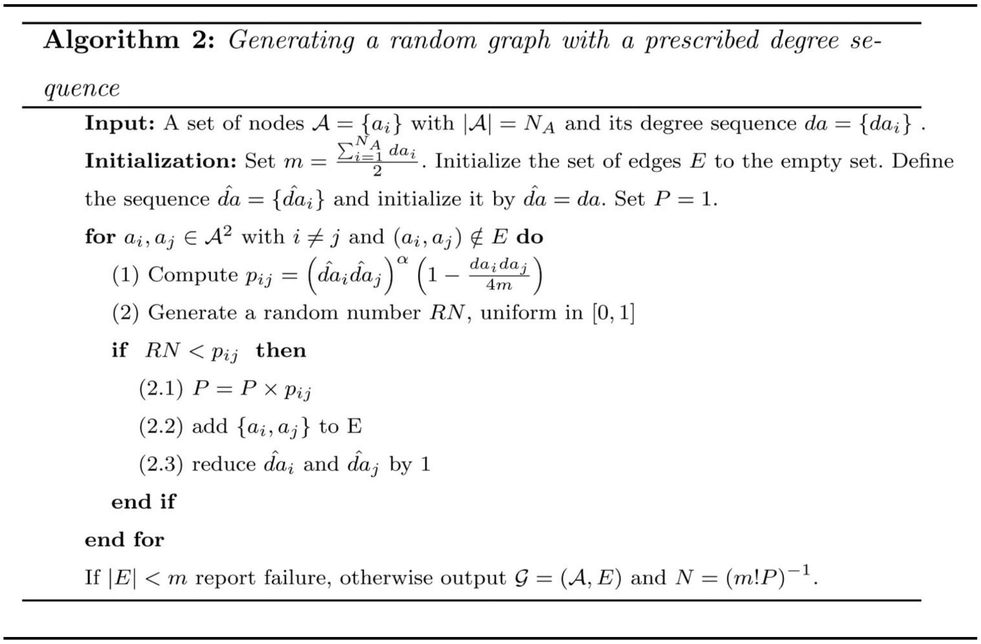


Fig. 7.

Two levels of the ultrametric geometry on the network of the world-wide major airports. (a) At a relatively low temperature, 12 major clusters are detected (a cluster is deemed major if it contains at least four airports). The marginal clusters (i.e. with less than four airports) are represented with black crosses. Flight routes are shown as red line just for illustration purpose; there are no indications on the frequency of ights along these routes; (b) At a higher temperature only six major clusters are observed. Clusters 1 and 3, and clusters 4, 5 and 6 observed at the lower temperate are merged at this level (see text for details). Color online.

Algorithm 1: *Optimizing the Parisi matrix under an Ising-like 2D potential*

```
Input: Network $\mathcal{G}$; Parisi adjacency matrix $A_P$ from DCG clustering on nodes.
    (1) Identify and label blocks within $A_P$ based on the the finest scale level of the DCG tree
    (2) Permute the blocks within $A_P$ identified in step (1) to decrease $E(A) \rightarrow A_P(U_1)$
    (3) Permute nodes in $A_P(U_1)$ within blocks to further minimize $E[A] \rightarrow A_P(U_2)$
    (4) Randomly select and switch two nodes from different blocks using a Metropolis-Hasting algorithm to explore potential reduction of the energy of $A_P(U_2) \rightarrow \hat{U}^*$ and $A_P(\hat{U}^*)$
```



Algorithm 2: *Generating a random graph with a prescribed degree sequence*

```
Input: A set of nodes $\mathcal{A} = \{a_i\}$ with $|\mathcal{A}| = N_A$ and its degree sequence $da = \{da_i\}$ .
Initialization: Set $m = \frac{\sum_{i=1}^{N_A} da_i}{2}$. Initialize the set of edges $E$ to the empty set. Define
the sequence $\hat{da} = \{\hat{da}_i\}$ and initialize it by $\hat{da} = da$. Set $P = 1$.
for $a_i, a_j \in \mathcal{A}^2$ with $i \neq j$ and $(a_i, a_j) \notin E$ do
    (1) Compute $p_{ij} = \left(\hat{da}_i \hat{da}_j\right)^{\alpha} \left(1 - \frac{da_i da_j}{4m}\right)$
    (2) Generate a random number $RN$, uniform in $[0, 1]$
    if $RN < p_{ij}$ then
        (2.1) $P = P \times p_{ij}$
        (2.2) add $\{a_i, a_j\}$ to E
        (2.3) reduce $\hat{da}_i$ and $\hat{da}_j$ by 1
    end if
end for
If $|E| < m$ report failure, otherwise output $\mathcal{G} = (\mathcal{A}, E)$ and $N = (m!P)^{-1}$.
```