

Hierarchical structure and the prediction of missing links in networks

Aaron Clauset,^{1,3} Cristopher Moore,^{1,2,3} M. E. J. Newman^{3,4*}

¹Department of Computer Science and ²Department of Physics and Astronomy,
University of New Mexico, Albuquerque, NM 87131, USA

³Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM 87501, USA

⁴Department of Physics and Center for the Study of Complex Systems,
University of Michigan, Ann Arbor, MI 48109, USA

*To whom correspondence should be addressed. E-mail: mejn@umich.edu.

Networks have in recent years emerged as an invaluable tool for describing and quantifying complex systems in many branches of science (1–3). Recent studies suggest that networks often exhibit hierarchical organization, where vertices divide into groups that further subdivide into groups of groups, and so forth over multiple scales. In many cases these groups are found to correspond to known functional units, such as ecological niches in food webs, modules in biochemical networks (protein interaction networks, metabolic networks, or genetic regulatory networks), or communities in social networks (4–7). Here we present a general technique for inferring hierarchical structure from network data and demonstrate that the existence of hierarchy can simultaneously explain and quantitatively reproduce many commonly observed topological properties of networks, such as right-skewed degree distributions, high clustering coefficients, and short path lengths. We further show that knowledge of hierarchical structure can be used to predict missing connections in partially known networks with high accuracy, and for more general network

structures than competing techniques (8). Taken together, our results suggest that hierarchy is a central organizing principle of complex networks, capable of offering insight into many network phenomena.

A great deal of recent work has been devoted to the study of clustering and community structure in networks (5, 6, 9–11). Hierarchical structure goes beyond simple clustering, however, by explicitly including organization at all scales in a network simultaneously. Conventionally, hierarchical structure is represented by a tree or *dendrogram* in which closely related pairs of vertices have lowest common ancestors that are lower in the tree than those of more distantly related pairs—see Fig. 1. We expect the probability of a connection between two vertices to depend on their degree of relatedness. Structure of this type can be modeled mathematically using a probabilistic approach in which we endow each internal node r of the dendrogram with a probability p_r and then connect each pair of vertices for whom r is the lowest common ancestor independently with probability p_r (Fig. 1c).

This model, which we call a *hierarchical random graph*, is similar in spirit (although different in realization) to the tree-based models used in some studies of network search and navigation (12, 13). Like most work on community structure, it assumes that communities at each level of organization are disjoint. Overlapping communities have occasionally been studied (see, for example, (14)) and could be represented using a more elaborate probabilistic model, but as we discuss below the present model already captures many of the structural features of interest.

Given a dendrogram and a set of probabilities p_r , the hierarchical random graph model allows us to generate artificial networks with a specified hierarchical structure, a procedure that might be useful in certain situations. Our goal here, however, is a different one. We would like to detect and analyze the hierarchical structure, if any, of networks in the real world. We accomplish this by fitting the hierarchical model to observed network data using the tools of statistical inference, combining a maximum likelihood approach (15) with a Monte Carlo

sampling algorithm (16) on the space of all possible dendrograms. This technique allows us to sample hierarchical random graphs with probability proportional to the likelihood that they generate the observed network. To obtain the results described below we combine information from a large number of such samples, each of which is a reasonably likely model of the data.

The success of this approach relies on the flexible nature of our hierarchical model, which allows us to fit a wide range of network structures. The traditional picture of communities or modules in a network, for example, corresponds to connections that are dense within groups of vertices and sparse between them—a behavior called “assortativity” in the literature (17). The hierarchical random graph can capture behavior of this kind using probabilities p_r that decrease as we move higher up the tree. Conversely, probabilities that increase up the tree correspond to “disassortative” structures in which vertices are less likely to be connected on small scales than on large ones. By letting the p_r values vary arbitrarily throughout the dendrogram, the hierarchical random graph can capture both assortative and disassortative structure, as well as arbitrary mixtures of the two, at all scales and in all parts of the network.

To demonstrate our method we have used it to construct hierarchical decompositions of three example networks drawn from disparate fields: the metabolic network of the spirochete *Treponema pallidum* (18), a network of associations between terrorists (19), and a food web of grassland species (20). To test whether these decompositions accurately capture the networks’ important structural features, we use the sampled dendrograms to generate new networks, different in detail from the originals but, by definition, having similar hierarchical structure (see the Supplementary Information for more details). We find that these “resampled” networks match the statistical properties of the originals quite closely, including degree distribution, clustering coefficient, and distribution of shortest path lengths between pairs of vertices, despite the fact that none of these properties is explicitly represented in the hierarchical random graph (Table 1 and Fig. 2a,b). Thus it appears that a network’s hierarchical structure is capable of explaining a

Network	$\langle k \rangle_{\text{real}}$	$\langle k \rangle_{\text{samp}}$	C_{real}	C_{samp}	d_{real}	d_{samp}
<i>T. pallidum</i>	4.8	3.7(1)	0.0625	0.0444(2)	3.690	3.940(6)
Terrorists	4.9	5.1(2)	0.361	0.352(1)	2.575	2.794(7)
Grassland	3.0	2.9(1)	0.174	0.168(1)	3.29	3.69(2)

Table 1: Comparison of network statistics for the three example networks studied and new networks generated by resampling from our hierarchical model. The generated networks closely match the average degree $\langle k \rangle$, clustering coefficient C , and average vertex-vertex distance d in each case, suggesting that they capture much of the real networks’ structure. Parenthetical values indicate standard errors on the final digits.

wide variety of other network features as well.

The dendrograms produced by our method are also of interest in themselves, as a graphical representation and summary of the hierarchical structure of the observed network. As discussed above, our method typically generates not just a single dendrogram but a set of dendrograms, each of which is a good fit to the data. From this set we can, using techniques from phylogeny reconstruction (21), create a single *consensus dendrogram*, which captures the topological features that appear consistently across all or a large fraction of the dendrograms and typically represents a better summary of the network’s structure than any individual dendrogram. Figure 2c shows such a consensus dendrogram for the grassland species network, which clearly reveals communities and sub-communities of plants, herbivores, parasitoids, and hyper-parasitoids.

Another application of the hierarchical decomposition is in the prediction of missing interactions in networks. In many settings, the discovery of interactions in a network requires significant experimental effort in the laboratory or the field. As a result, our current pictures of many networks are substantially incomplete (22–29). An attractive alternative to checking exhaustively for a connection between every pair of vertices in a network is to try to predict, in advance and based on the connections already observed, which vertices are most likely to be connected, so that scarce experimental resources can be focused on testing for those interactions. If our predictions are good, we can in this way reduce substantially the effort required to

establish the network’s topology.

The hierarchical decomposition can be used as the basis for an effective method of predicting missing interactions as follows. Given an observed but incomplete network, we generate as described above a set of hierarchical random graphs—dendrograms and the associated probabilities p_r —that fit that network. Then we look for pairs of vertices that have a high average probability of connection within these hierarchical random graphs but which are unconnected in the observed network. These pairs we consider the most likely candidates for missing connections. (Technical details of the procedure are given in the Supplementary Information.)

We demonstrate the method using our three example networks again. For each network we remove a subset of connections chosen uniformly at random and then attempt to predict, based on the remaining connections, which ones have been removed. A standard metric for quantifying the accuracy of prediction algorithms, commonly used in the medical sciences and machine learning communities, is the AUC statistic, which is equivalent to the area under the receiver-operating characteristic (ROC) curve (see, for example, (30)). In the present context, the AUC statistic can be interpreted as the probability that a randomly chosen missing connection (a true positive) is given a higher score by our method than a randomly chosen pair of unconnected vertices (a true negative). Thus, the degree to which the AUC exceeds 0.5 indicates how much better our predictions are than chance. Figure 3 shows the AUC statistic for the three networks as a function of the fraction of the connections known to the algorithm. For all three networks our algorithm does far better than chance, indicating that hierarchy is a strong general predictor of missing structure.

It is also instructive to compare the performance of our method to that of other methods for link prediction (8). Previously proposed methods include assuming that vertices are likely to be connected if they have many common neighbors, if there are short paths between them, or if the product of their degrees is large. These approaches work well for strongly assortative networks

such as the collaboration and citation networks studied in (8) and for the metabolic and terrorist networks studied here (Fig. 3a,b). Indeed, for the metabolic network, the shortest-path heuristic performs better than our algorithm.

However, these simple methods can be misleading for networks that exhibit more general types of structure. In food webs, for instance, pairs of predators often share prey species, but rarely prey on each other. In such situations a common-neighbor or shortest-path-based method would predict connections between predators where none exist. The hierarchical model, by contrast, is capable of expressing both assortative and disassortative structure and, as Fig. 3c shows, gives substantially better predictions for the grassland network. (Indeed, in Fig. 2d there are several groups of parasitoids that our algorithm has grouped together in a disassortative community, in which they prey on the same herbivore but not on each other.) The hierarchical method thus makes accurate predictions for a wider range of network structures than the alternative methods above.

In the applications above, we have assumed for simplicity that there are no false positives in our network data, i.e., that every observed edge corresponds to a real interaction. In networks where false positives may be present, however, they too could be predicted using the same approach: we would simply look for pairs of vertices that have a *low* average probability of connection within the hierarchical random graph but which are connected in the observed network.

The method described here could also be extended to incorporate domain-specific information, such as morphological or behavioral traits in food webs (29) or phylogenetic or binding-domain data for biochemical networks (23), by adjusting the probabilities of edges accordingly. As the results above show, however, we can obtain good predictions even in the absence of such information, indicating that topology alone can provide rich insights.

In closing, we note that our approach differs crucially from previous work on hierarchical

structure in networks (1, 4–7, 9, 11, 31) in that it acknowledges explicitly that most real-world networks have many plausible hierarchical representations of roughly equal likelihood. Previous work, by contrast, has typically sought a single hierarchical representation for a given network. By sampling an ensemble of dendrograms, our approach avoids over-fitting the data and allows us to explain many common topological features, generate resampled networks with similar structure to the original, derive a clear and concise summary of a network’s structure via its consensus dendrogram, and accurately predict missing connections in a wide variety of situations.

Acknowledgments: The authors thank Chris Wiggins, Cosma Shalizi, Jennifer Dunne, Michael Gastner, Mason Porter, Petter Holme, and Mikael Huss for their help. CM thanks the Center for the Study of Complex Systems at the University of Michigan for their hospitality while some of this work was conducted. This work was supported by the National Science Foundation under grants PHY–0200909 (AC, CM), ITR–0324845 (AC, CM) and DMS–0405348 (MEJN), by a grant from the James S. McDonnell Foundation (MEJN), and by the Santa Fe Institute (AC, CM, MEJN).

References

1. S. Wasserman, K. Faust, *Social Network Analysis* (Cambridge University Press, Cambridge, 1994).
2. R. Albert, A.-L. Barabási, *Rev. Mod. Phys.* **74**, 47 (2002).
3. M. E. J. Newman, *SIAM Review* **45**, 167 (2003).
4. E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, A.-L. Barabási, *Science* **30**, 1551 (2002).

5. A. Clauset, M. E. J. Newman, C. Moore, *Phys. Rev. E* **70**, 066111 (2004).
6. R. Guimera, L. A. N. Amaral, *Nature* **433**, 895 (2005).
7. M. C. Lagomarsino, P. Jona, B. Bassetti, H. Isambert, *Proc. Natl. Acad. Sci. USA* **104**, 5516 (2001).
8. D. Liben-Nowell, J. Kleinberg, *International Conference on Information and Knowledge Management* (2003). It should be noted that this paper focuses on predicting future connections given current ones, as opposed to predicting connections which exist but have not yet been observed.
9. M. Girvan, M. E. J. Newman, *Proc. Natl. Acad. Sci. USA* **99**, 7821 (2002).
10. A. E. Krause, K. A. Frank, D. M. Mason, R. E. Ulanowicz, W. W. Taylor, *Nature* **426**, 282 (2003).
11. F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, *Proc. Natl. Acad. Sci. USA* **101**, 2658 (2004).
12. D. J. Watts, P. S. Dodds, M. E. J. Newman, *Science* **296**, 1302 (2002).
13. J. M. Kleinberg, *Proceedings of the 2001 Neural Information Processing Systems Conference*, T. G. Dietterich, S. Becker, Z. Ghahramani, eds. (MIT Press, Cambridge, MA, 2002).
14. G. Palla, I. Derényi, I. Farkas, T. Vicsek, *Nature* **435**, 814 (2005).
15. G. Casella, R. L. Berger, *Statistical Inference* (Duxbury Press, Belmont, 2001).
16. M. E. J. Newman, G. T. Barkema, *Monte Carlo Methods in Statistical Physics* (Clarendon Press, Oxford, 1999).

17. M. E. J. Newman, *Phys. Rev. Lett.* **89**, 208701 (2002).
18. M. Huss, P. Holme, Currency and commodity metabolites: Their identification and relation to the modularity of metabolic networks. (2006). Preprint `q-bio.MN/0603038`.
19. V. Krebs, *Connections* **24**, 43 (2002).
20. H. A. Dawah, B. A. Hawkins, M. F. Claridge, *Journal of Animal Ecology* **64**, 708 (1995).
21. D. Bryant, *BioConsensus*, M. Janowitz, F.-J. Lapointe, F. R. McMorris, B. Mirkin, F. Roberts, eds. (DIMACS, 2003), pp. 163–184.
22. J. A. Dunne, R. J. Williams, N. D. Martinez, *Proc. Natl. Acad. Sci. USA* **99**, 12917 (2002).
23. A. Szilágyi, V. Grimm, A. K. Arakaki, J. Skolnick, *Physical Biology* **2**, S1 (2005).
24. E. Sprinzak, S. Sattath, H. Margalit, *J. Mol. Biol.* **327**, 919 (2003).
25. T. Ito, *et al.*, *Proc. Natl. Acad. Sci. USA* **98**, 4569 (2001).
26. A. Lakhina, J. W. Byers, M. Crovella, P. Xie, *Proc. INFOCOM* (2003).
27. A. Clauset, C. Moore, *Phys. Rev. Lett.* **94**, 018701 (2005).
28. D. Achlioptas, A. Clauset, D. Kempe, C. Moore, *Proc. Symposium on Theory of Computing (STOC)* (2005), pp. 694–703.
29. N. D. Martinez, B. A. Hawkins, H. A. Dawah, B. P. Feifarek, *Ecology* **80**, 1044 (1999).
30. J. A. Hanel, B. J. McNeil, *Radiology* **143**, 29 (1982).
31. M. Sales-Pardo, R. Guimerá, A. A. Moreira, L. A. N. Amaral, Extracting and representing the hierarchical organization of complex systems (2006). Unpublished manuscript.

Supplementary Information

A Hierarchical Random Graphs

Our model for the hierarchical organization of a network is as follows. Let G be a graph with n vertices. A *dendrogram* D is a binary tree with n leaves corresponding to the vertices of G . Each of the $n - 1$ internal nodes of D corresponds to a group of vertices that are descended from it. We associate a probability p_r with each internal node r . Then, given two vertices i, j of G , the probability p_{ij} that they are connected by an edge is $p_{ij} = p_r$ where r is their lowest common ancestor in D . The combination $(D, \{p_r\})$ of the dendrogram and the set of probabilities then defines a *hierarchical random graph*.

Note that if a community has, say, three subcommunities, with an equal probability p of connections between them, we can represent this in our model by first splitting one of these subcommunities off, and then splitting the other two. The two internal nodes corresponding to these splits would be given the same probabilities $p_r = p$. This yields three possible binary dendrograms, which are all considered equally likely.

We can think of the hierarchical random graph as a variation on the classical Erdős–Rényi random graph $G(n, p)$. As in that model, the presence or absence of an edge between any pair of vertices is independent of the presence or absence of any other edge. However, whereas in $G(n, p)$ every pair of vertices has the same probability p of being connected, in the hierarchical random graph the probabilities are inhomogeneous, with the inhomogeneities controlled by the topological structure of the dendrogram D and the parameters $\{p_r\}$. Many other models with inhomogeneous edge probabilities have, of course, been studied in the past. Examples include the widely-studied configuration model¹ and structured random graphs in which there are a finite number of types of vertices with a matrix p_{kl} giving the connection probabilities

¹M. Molloy and B. Reed, “A critical point for random graphs with a given degree sequence”, *Random Structures and Algorithms* **6**, 161–179 (1995)

between them.²

B Fitting the hierarchical random graph to data

Now we turn to the question of finding the hierarchical random graph or graphs that best fits the observed real-world network G . Assuming that all hierarchical random graphs are *a priori* equally likely, the probability that a given model $(D, \{p_r\})$ is the correct explanation of the data is, by Bayes' theorem, proportional to the posterior probability or *likelihood* \mathcal{L} with which that model generates the observed network.³ Our goal is to maximize \mathcal{L} or, more generally, to sample with probability proportional to \mathcal{L} from the space of all models.

Let E_r be the number of edges in G whose endpoints have r as their lowest common ancestor in D , and let L_r and R_r , respectively, be the numbers of leaves in the left and right subtrees rooted at r . Then the likelihood of the hierarchical random graph (consisting of the dendrogram D and the set of probabilities $\{p_r\}$) is

$$\mathcal{L}(D, \{p_r\}) = \prod_{r \in D} p_r^{E_r} (1 - p_r)^{L_r R_r - E_r} \quad (1)$$

with the convention that $0^0 = 1$.

If we fix the dendrogram D , it is easy to find the probabilities $\{\bar{p}_r\}$ that maximize $\mathcal{L}(D, \{p_r\})$. For each r , they are given by

$$\bar{p}_r = \frac{E_r}{L_r R_r}, \quad (2)$$

the fraction of potential edges between the two subtrees of r that actually appear in the graph G .

The likelihood of the dendrogram evaluated at the maximum is then

$$\mathcal{L}(D) = \prod_{r \in D} \left[\bar{p}_r^{\bar{p}_r} (1 - \bar{p}_r)^{1 - \bar{p}_r} \right]^{L_r R_r}. \quad (3)$$

²F. McSherry, "Spectral Partitioning of Random Graphs," *Proc. Foundations of Computer Science (FOCS)*, pp. 529–537 (2001)

³G. Casella and R. L. Berger, "Statistical Inference." Duxbury Press, Belmont (2001).

Figure 4 shows an illustrative example, consisting of a network with six vertices.

It is often convenient to work with the logarithm of the likelihood,

$$\log \mathcal{L}(D) = - \sum_{r \in D} L_r R_r h(\bar{p}_r), \quad (4)$$

where $h(p) = -p \log p - (1 - p) \log(1 - p)$ is the Gibbs-Shannon entropy function. Note that each term $-L_r R_r h(\bar{p}_r)$ is maximized when \bar{p}_r is close to 0 or to 1, i.e., when the entropy is minimized. In other words, high-likelihood dendrograms are those that partition the vertices into groups between which connections are either very common or very rare.

We now use a Markov chain Monte Carlo method to sample dendrograms D with probability proportional to their likelihood $\mathcal{L}(D)$. To create the Markov chain we need to pick a set of transitions between possible dendrograms. The transitions we use consist of rearrangements of subtrees of the dendrogram as follows. First, note that each internal node r of a dendrogram D is associated with three subtrees: the subtrees s, t descended from its two daughters, and the subtree u descended from its sibling. As Figure 5 shows, there are two ways we can reorder these subtrees without disturbing any of their internal relationships. Each step of our Markov chain consists first of choosing an internal node r uniformly at random (other than the root) and then choosing uniformly at random between the two alternate configurations of the subtrees associated with that node and adopting that configuration. The result is a new dendrogram D' . It is straightforward to show that transitions of this type are *ergodic*, i.e., that any pair of finite dendrograms can be connected by a finite series of such transitions.

Once we have generated our new dendrogram D' we accept or reject that dendrogram according to the standard Metropolis–Hastings rule (16). Specifically, we accept the transition $D \rightarrow D'$ if $\Delta \log \mathcal{L} = \log \mathcal{L}(D') - \log \mathcal{L}(D)$ is nonnegative, so that D' is at least as likely as D ; otherwise we accept the transition with probability $\exp(\log \Delta \mathcal{L}) = \mathcal{L}(D')/\mathcal{L}(D)$. If the transition is not accepted we revert to the original dendrogram D again on this step of the chain.

The Metropolis-Hastings rule ensures detailed balance and, in combination with the ergodic transition set, guarantees a limiting probability distribution over dendrograms that is proportional to the likelihood, $P(D) \propto \mathcal{L}(D)$. The quantity $\Delta \log \mathcal{L}$ can be calculated easily, since the only terms in Eq. (4) that change from D to D' are those involving the subtrees s , t , and u associated with the chosen node.

The Markov chain appears to converge relatively quickly, with the likelihood reaching a plateau after roughly $O(n^2)$ steps. This is not a rigorous performance guarantee, however, and indeed there are mathematical results for similar Markov chains that suggest that equilibration could take exponential time in the worst case.⁴ Still, as our results here show, the method seems to work quite well in practice. The algorithm is able to handle networks with up to a few thousand vertices in a reasonable amount of computer time.

We find that there are typically many dendrograms with roughly competitive likelihoods, which reinforces our contention that it is important to sample the distribution of dendrograms rather than merely focusing on the most likely one.

C Resampling from the hierarchical random graph

The procedure for resampling from the hierarchical random graph is as follows.

1. Initialize the Markov chain by choosing a random starting dendrogram.
2. Run the Monte Carlo algorithm until equilibrium is reached.
3. Sample dendrograms at regular intervals thereafter from those generated by the Markov chain.

⁴E. Mossel and E. Vigoda, “Phylogenetic MCMC Are Misleading on Mixtures of Trees.” *Science* **309**, 2207 (2005)

4. For each sampled dendrogram D , create a resampled graph G' with n vertices by placing an edge between each of the $n(n - 1)/2$ vertex pairs (i, j) with independent probability \bar{p}_r , where r is the lowest common ancestor of i and j in D and \bar{p}_r is given by Eq. (2). (In principle, there is nothing to stop us generating many resampled graphs from a dendrogram, but in the calculations described in this paper we generate only one from each dendrogram.)

After generating many samples in this way, we can compute averages of network statistics such as the degree distribution, the clustering coefficient, the vertex-vertex distance distribution, and so forth. Thus, in a way similar to Bayesian model averaging,⁵ we can estimate the distribution of network statistics defined by the equilibrium ensemble of dendrograms.

For the construction of consensus dendrograms such as the one shown in Fig. 2c, we found it useful to weight the most likely dendrograms more heavily, giving them weight proportional to the square of their likelihood, in order to extract a coherent consensus structure from the equilibrium set of models.

D Predicting Missing Connections

Our algorithm for using hierarchical random graphs to predict missing connections is as follows.

1. Initialize the Markov chain by choosing a random starting dendrogram.
2. Run the Monte Carlo algorithm until equilibrium is reached.
3. Sample dendrograms at regular intervals thereafter from those generated by the Markov chain.

⁵T. Hastie, R. Tibshirani and J. Friedman, “The Elements of Statistical Learning.” Springer, New York (2001).

4. For each pair of vertices i, j for which there is not already a known connection, calculate the mean probability $\langle p_{ij} \rangle$ that they are connected by averaging over the corresponding probabilities p_{ij} in each of the sampled dendrograms D .
5. Sort these pairs i, j in decreasing order of $\langle p_{ij} \rangle$ and predict that the highest ranked ones have missing connections.

In general, we find that the top 1% of such predictions are highly accurate. However, for large networks, even the top 1% can be an unreasonably large number of candidates to check experimentally. In many contexts, researchers may want to consider using the procedure interactively, i.e., predicting a small number of missing connections, checking them experimentally, adding the results to the network, and running the algorithm again to predict additional connections.

The alternative prediction methods we compared against, which were previously investigated in (8), consist of giving each pair i, j of vertices a score, sorting pairs in decreasing order of their score, and predicting that those with the highest scores are the most likely to be connected. Several different types of scores were investigated, defined as follows, where $\Gamma(j)$ is the set of vertices connected to j .

1. Common neighbors: $\text{score}(i, j) = |\Gamma(i) \cap \Gamma(j)|$, the number of common neighbors of vertices i and j .
2. Jaccard coefficient: $\text{score}(i, j) = |\Gamma(i) \cap \Gamma(j)| / |\Gamma(i) \cup \Gamma(j)|$, the fraction of all neighbors of i and j that are common neighbors of both.
3. Degree product: $\text{score}(i, j) = |\Gamma(i)| |\Gamma(j)|$, the product of the degrees of i and j .
4. Short paths: $\text{score}(i, j)$ is 1 divided by the length of the shortest path through the network from i to j (or zero for vertex pairs that are not connected by any path).

One way to quantify the success of a prediction method, used by previous authors who have studied link prediction problems (8), is the ratio between the probability that the top-ranked pair is connected and the probability that a randomly chosen pair of vertices, which do not have an observed connection between them, are connected. Figure 6 shows the average value of this ratio as a function of the percentage of the network shown to the algorithm, for each of our three networks. Even when fully 50% of the network is missing, our method predicts missing connections about ten times better than chance for all three networks. In practical terms, this means that the amount of work required of the experimenter to discover a new connection is reduced by a factor of 10, an enormous improvement by any standard. If a greater fraction of the network is known, the accuracy becomes even greater, rising as high as 200 times better than chance when only a few connections are missing.

We note, however, that this using this ratio to judge prediction algorithms has an important disadvantage. Some missing connections are much easier to predict than others: for instance, if a network has a heavy-tailed degree distribution and we remove a randomly chosen subset of the edges, the chances are excellent that two high-degree vertices will have a missing connection and such a connection can be easily predicted by simple heuristics such as those discussed above. The AUC statistic used in the text, by contrast, looks at an algorithm’s overall ability to rank all the missing connections over nonexistent ones, not just those that are easiest to predict.

Finally, we have investigated the performance of each of the prediction algorithms on purely random (i.e., Erdős–Rényi) graphs. As expected, no method performs better than chance in this case, since the connections are completely independent random events and there is no structure to discover. We also tested each algorithm on a graph with a power-law degree distribution generated according to the configuration model. In this case, guessing that high-degree vertices are likely to be connected performs quite well, whereas the method based on the hierarchical random graph performs poorly since these graphs have no hierarchical structure to discover.

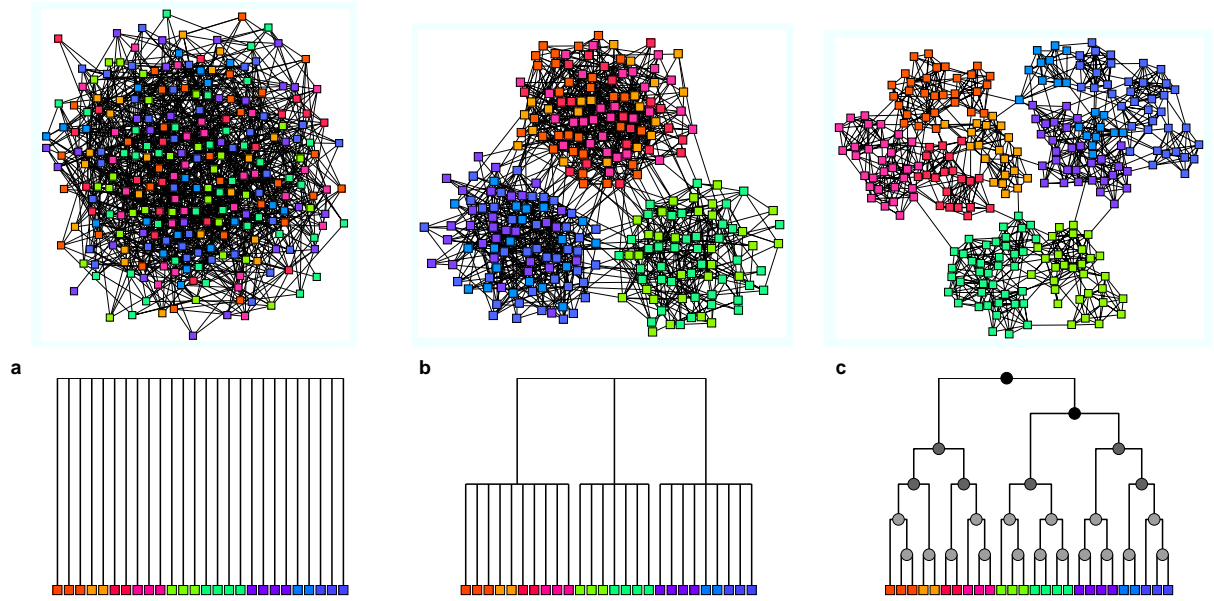


Figure 1: **a**, A simple network with no hierarchical structure and the corresponding dendrogram. **b**, A network with a single level of community structure and its dendrogram. **c**, A hierarchical network with structure on many scales and the corresponding hierarchical random graph. In the hierarchical random graph the dendrogram structure is augmented by defining for each internal node r a probability p_r that a pair of vertices in the left and right subtrees of that node are connected. (The shades of the internal nodes in the figure represent the probabilities.)

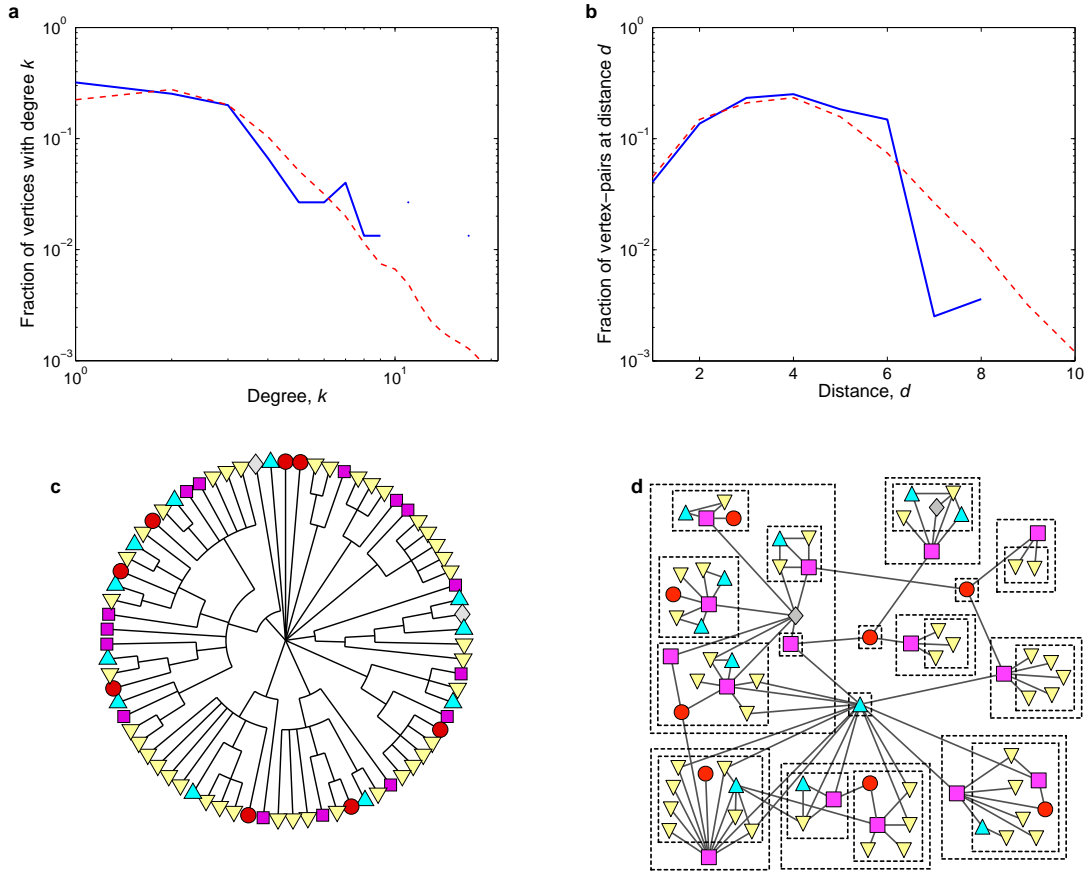


Figure 2: Application of our hierarchical decomposition to the network of grassland species interactions. **a**, Original (blue) and resampled (red) degree distributions. **b**, Original and resampled distributions of vertex-vertex distances. **c**, Consensus dendrogram reconstructed from the sampled hierarchical models. **d**, A visualization of the network in which the upper few levels of the consensus dendrogram are shown as boxes around species (plants \circ , herbivores \square , parasitoids ∇ , hyper-parasitoids \triangle and hyper-hyper-parasitoids \diamond). Note that in several cases, a set of parasitoids is grouped into a community by our algorithm, not because they prey on each other, but because they prey on the same herbivore.

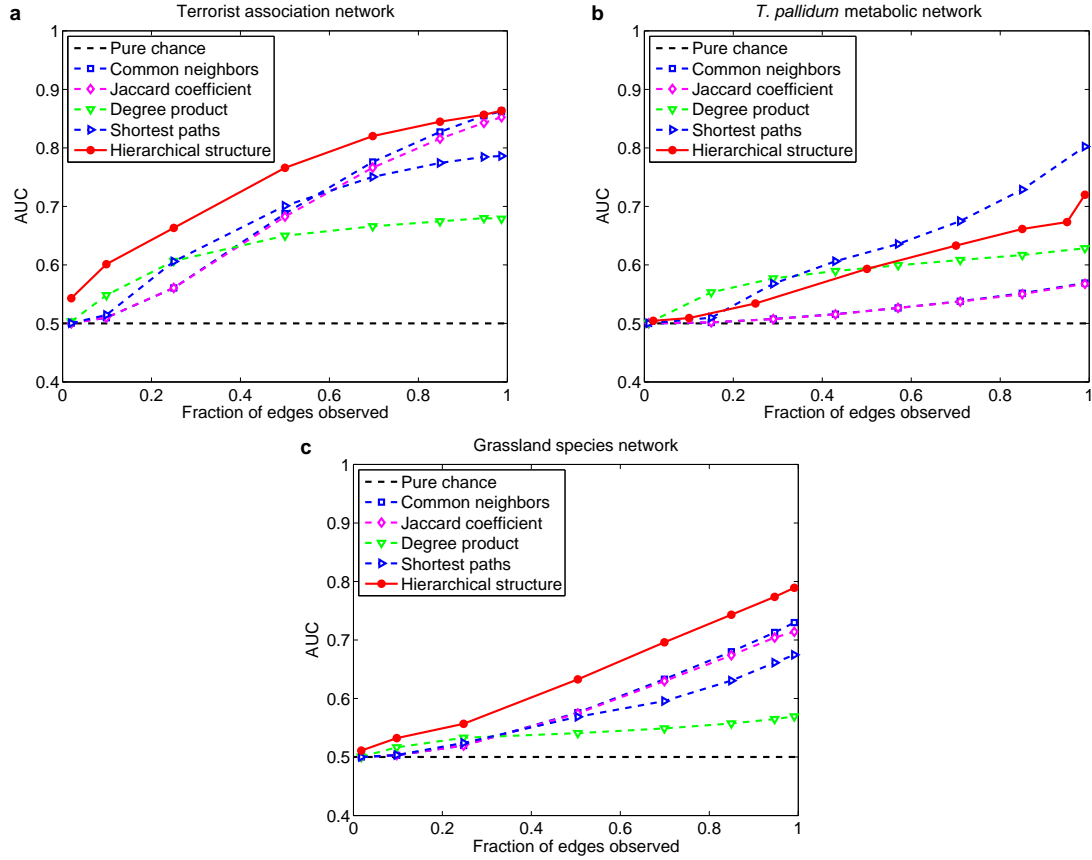


Figure 3: Comparison of link prediction methods. Average area under the curve (AUC) statistic for the receiver-operating characteristic (ROC), as a function of the fraction of connections known to the algorithm, for the link prediction method presented here and a variety of previously published methods.

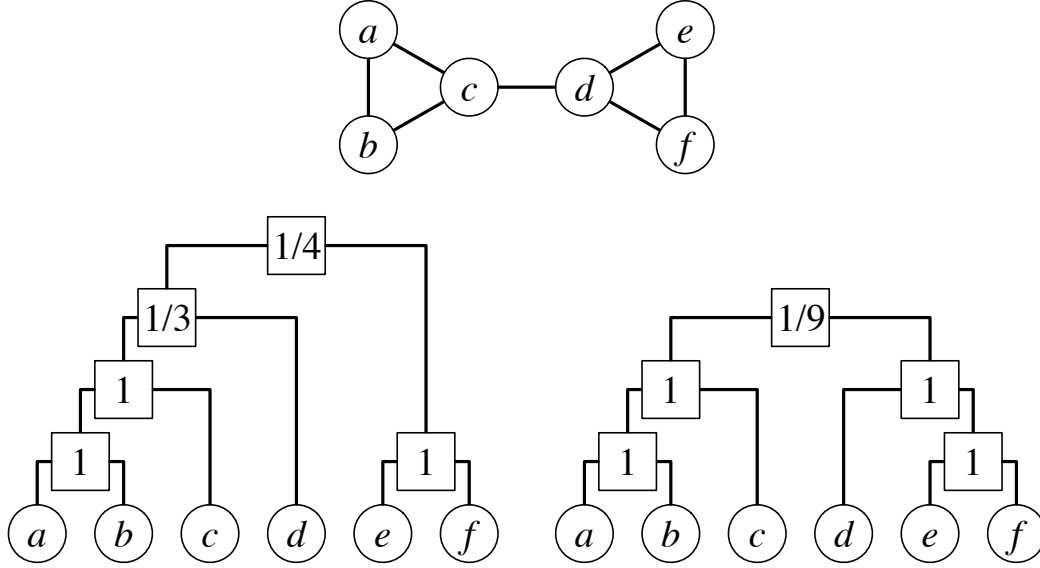


Figure 4: An example network G consisting of six vertices, and the likelihood of two possible dendrograms. The internal nodes r of each dendrogram are labeled with the maximum-likelihood probability \bar{p}_r , i.e., the fraction of potential edges between their left and right subtrees that exist in G . According to Eq. (3), the likelihoods of the two dendrograms are $\mathcal{L}(D_1) = (1/3)(2/3)^2 \cdot (1/4)^2(3/4)^6 = 0.00165\dots$ and $\mathcal{L}(D_2) = (1/9)(8/9)^8 = 0.0433\dots$. The second dendrogram is far more likely because it correctly divides the network into two highly-connected subgraphs at the first level.

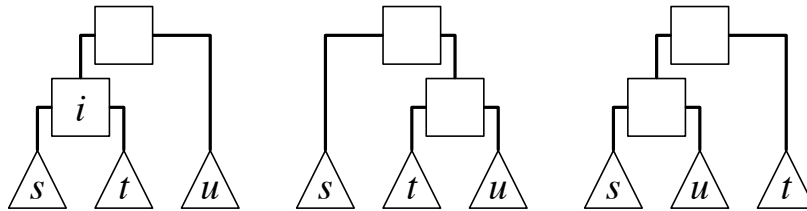


Figure 5: Each internal node r of the dendrogram has three associated subtrees s , t , and u , which can be placed in any of three configurations. (Note that the topology of the dendrogram depends only on the sibling and parent relationships; the order, left to right, in which they are depicted is irrelevant).

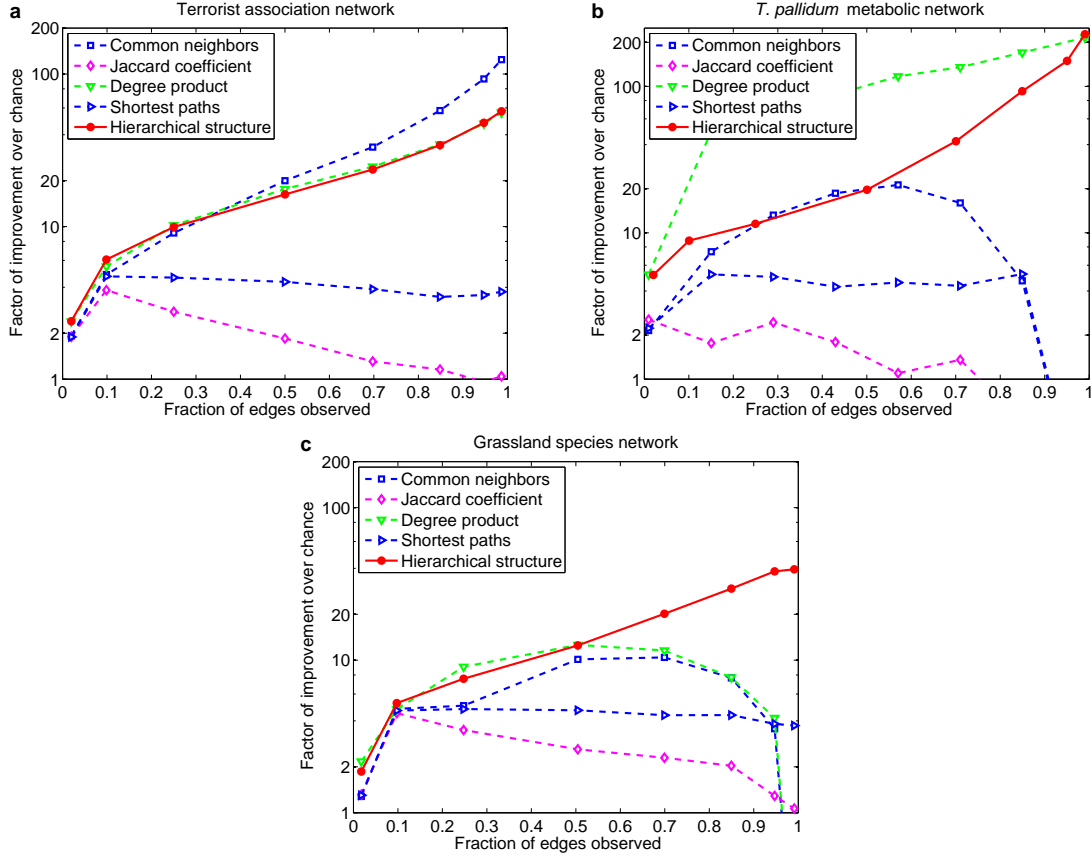


Figure 6: Further comparison of link prediction algorithms. Data points represent the average ratio between the probability that the top-ranked pair of vertices is in fact connected and the corresponding probability for a randomly-chosen pair, as a function of the fraction of the connections known to the algorithm. For each network (**a**, Terrorist associations; **b**, *T. pallidum* metabolites; and **c**, Grassland species interactions), we compare our method with simpler methods such as guessing that two vertices are connected if they share common neighbors, have a high degree product, or have a short path between them.