Topology Discovery of Sparse Random Graphs With Few Participants*

Animashree Anandkumar†, Avinatan Hassidim‡and Jonathan Kelner§

February 28, 2012

Abstract

We consider the task of topology discovery of sparse random graphs using end-to-end random measurements (e.g., delay) between a subset of nodes, referred to as the participants. The rest of the nodes are hidden, and do not provide any information for topology discovery. We consider topology discovery under two routing models: (a) the participants exchange messages along the shortest paths and obtain end-to-end measurements, and (b) additionally, the participants exchange messages along the second shortest path. For scenario (a), our proposed algorithm results in a sub-linear edit-distance guarantee using a sub-linear number of uniformly selected participants. For scenario (b), we obtain a much stronger result, and show that we can achieve consistent reconstruction when a sub-linear number of uniformly selected nodes participate. This implies that accurate discovery of sparse random graphs is tractable using an extremely small number of participants. We finally obtain a lower bound on the number of participants required by any algorithm to reconstruct the original random graph up to a given edit distance. We also demonstrate that while consistent discovery is tractable for sparse random graphs using a small number of participants, in general, there are graphs which cannot be discovered by any algorithm even with a significant number of participants, and with the availability of end-to-end information along all the paths between the participants.

Keywords: Topology Discovery, Sparse Random Graphs, End-to-end Measurements, Hidden Nodes, Quartet Tests.

1 Introduction

Inference of global characteristics of large networks using limited local information is an important and a challenging task. The discovery of the underlying network topology is one of the main goals of network inference, and its knowledge is crucial for many applications. For instance, in communication networks, many network monitoring applications rely on the knowledge of the routing topology, e.g., to evaluate the resilience of the network to failures [2, 3]; for network traffic

*A shorter version appears in [1]. This version is scheduled to appear in Journal on Random Structures and Algorithms.

†A. Anandkumar is with the Center for Pervasive Communications and Computing, Electrical Engineering and Computer Science Dept., University of California, Irvine, USA 92697. Email: a.anandkumar@uci.edu

‡A. Hassidim is with Google Research, Tel Aviv, Israel. Email: avinatanh@gmail.com

§J. Kelner is with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge MA 02139. Email: kelner@mit.edu

prediction [4, 5] and monitoring [6], anomaly detection [7], or to infer the sources of viruses and rumors in the network [8]. In the context of social networks, the knowledge of topology is useful for inferring many characteristics such as identification of hierarchy and community structure [9], prediction of information flow [10, 11], or to evaluate the possibility of information leakage from anonymized social networks [12].

Traditionally, inference of routing topology in communication networks has relied on tools such as traceroute and mtrace [13] to generate path information between a subset of nodes. However, these tools require cooperation of intermediate nodes or routers to generate messages using the Internal Control Message Protocol (ICMP). Increasingly, today many routers block traceroute requests due to privacy and security concerns [14, 15], there by making inference of topology using traceroute inaccurate. Moreover, traceroute requests are not scalable for large networks, and cannot discover layer-2 switches and MPLS (Multi-protocol Label Switching) paths, which are increasingly being deployed [16].

The alternative approach for topology discovery is the approach of *network tomography*. Here, topology inference is carried out from end-to-end packet probing measurements (e.g., delay) between a subset of nodes, without the need for cooperation between the intermediate (i.e., non-participating) nodes in the network. Due to its flexibility, such approaches are gaining increasing popularity (see Section 1.2 for details).

The approach of topology discovery using end-to-end measurements is also applicable in the context of social networks. In many social networks, some nodes may be unwilling to participate or cooperate with other nodes for discovering the network topology, and there may be many hidden nodes in "hard to reach" places of the network, e.g., populations of drug users, and so on. Moreover, in many networks, there may be a cost to probing nodes for information, e.g., when there is a cash reward offered for filling out surveys. For such networks, it is desirable to design algorithms which can discover the overall network topology using small fraction of participants who are willing to provide information for topology discovery.

There are many challenges to topology discovery. The algorithms need to be computationally efficient and provide accurate reconstruction using a small fraction of participating nodes. Moreover, inference of large topologies is a task of *high-dimensional learning* [17]. In such scenarios, typically, only a small number of end-to-end measurements are available relative to the size of the network to be inferred. It is desirable to have algorithms with low *sample complexity* (see Definition 3), where the number of measurements required to achieve a certain level of accuracy scales favorably with the network size.

It is indeed not tractable to achieve all the above objectives for discovery of general network topologies using an arbitrary set of participants. There are fundamental identifiability issues, and in general, no algorithm will be able to discover the underlying topology. We demonstrate this phenomenon in Section 8.2, where we construct a small network with a significant fraction of participants which suffers from non-identifiability. Instead, it is desirable to design topology discovery algorithms which have guaranteed performance for certain classes of graphs.

We consider the class of Erdős-Rényi random graphs [18]. These are perhaps the simplest as well as the most well-studied class of random graphs. Such random graphs can provide a reasonable explanation for peer-to-peer networks [19] and social networks [20]. We address the following issues in this paper: can we discover random graphs using a small fraction of participating nodes, selected uniformly at random? can we design efficient algorithms with low sample complexity and with provable performance guarantees? what kinds of end-to-end measurements between the

participants are useful for topology discovery? finally, given a set of participants, is there a lower bound on the error (edit distance) of topology discovery that is achievable by any algorithm? Our work addresses these questions and also provides insights into many complex issues involved in topology discovery.

1.1 Summary of Contributions

We consider the problem of topology discovery of sparse random graphs using a uniformly selected set of participants. Our contributions in this paper are three fold. First, we design an algorithm with provable performance guarantees, when only minimal end-to-end information between the participants is available. Second, we consider the scenario with additional information, and design a discovery algorithm with much better reconstruction guarantees. Third, we provide a lower bound on the edit distance of the reconstructed graph by any algorithm, for a given number of participants. Our analysis shows that random graphs can be discovered accurately and efficiently using an extremely small number of participants.

We consider reconstruction of the giant component of the sparse random graph up to its minimal representation, where there are no redundant hidden nodes (see Section 3.1). Our end-to-end measurement model consists of random samples (e.g., delay) along the shortest paths between the participants. Using these samples, we design the first random-discovery algorithm, referred to as the RGD1 algorithm, which performs local tests over small groups of participating nodes (known as the *quartet tests*), and iteratively merges them with the previously constructed structure. Such tests are known to be accurate for tree topologies [21], but have not been previously analyzed for random-graph topologies. We provide a sub-linear edit-distance guarantee (in the number of nodes) under RGD1 when there are roughly $n^{5/6}$ participants, where n is the number of nodes in the network. The algorithm is also simple to implement, and is computationally efficient.

We then extend the algorithm to the scenario where additionally, there are end-to-end measurements available along the second shortest paths between the participating nodes. Such information is available since nodes typically maintain information about alternative routing paths, should the shortest path fail. In this scenario, our algorithm RGD2, has a drastic improvement in accuracy under the same set of participating nodes. Specifically, we demonstrate that consistent discovery can be achieved under RGD2 algorithm when there are roughly $n^{11/12}$ number of participants, where n is the network size. Thus, we can achieve accurate topology discovery of random graphs using an extremely small number of participants. For both our algorithms, the sample complexity is poly-logarithmic in the network size, meaning that the number of end-to-end measurement samples needs to scale poly-logarithmically in the network size to obtain the stated edit-distance guarantees.

Our analysis in this paper thus reveals that sparse random graphs can be efficiently discovered using a small number of participants. Our algorithms exploit the *locally tree-like* property of random graphs [18], meaning that these graphs contain a small number of short cycles. This enables us to provide performance guarantees for quartet tests which are known to be accurate for tree topologies, and this is done by carefully controlling the distances used by the quartet tests. At the same time, we exploit the presence of cycles in random graphs to obtain much better guarantees than in the case of tree topologies. In other words, while tree topologies require participation of at least half the number of nodes (i.e., the leaves) for accurate discovery, random-graph topologies can be accurately discovered using a sub-linear number of participants.

Finally, we provide lower bounds on the reconstruction error under any algorithm for a given number of participants. Specifically, we show that if less than roughly $\sqrt{n}$ nodes participate in

topology discovery, reconstruction is impossible under any algorithm, where n is the network size. We also discuss topology discovery in general networks, and demonstrate identifiability issues involved in the discovery process. We construct a small network with a significant fraction of nodes as participants which cannot be reconstructed using end-to-end information on all possible paths between the participants. This is in contrast to random graphs, where consistent and efficient topology discovery is possible using a small number of participants.

To the best of our knowledge, this is the first work to undertake a systematic study of random-graph discovery using end-to-end measurements between a subset of nodes. Although we limit ourselves to the study of random graphs, our algorithms are based on the locally tree-like property, and are thus equally applicable for discovering other locally tree-like graphs such as the d-regular graphs and the *scale-free* graphs; the latter class is known to be a good model for social networks [20, 22] and peer-to-peer networks [19]. Indeed more sophisticated and general models for networks have been developed [23–25], but we defer their study for future work.

1.2 Related Work

Network tomography has been extensively studied in the past and various heuristics and algorithms have been proposed along with experimental results on real data. For instance, the area of mapping the internet topology is very rich and extensive, e.g., see [4,26–32]. In the context of social networks, the work in [33] considers prediction of positive and negative links, the work in [34] considers inferring networks of diffusion and influence and the work in [35] considers inferring latent social networks through spread of contagions. A wide range of network tomography solutions have been proposed for general networks. See [36] for a survey.

Topology discovery is an important component of network tomography. There have been several theoretical developments on this topic. The work in [37] provides hardness results for topology discovery under various settings. Topology discovery under availability of different kinds of queries have been previously considered, such as:
(i) Shortest-path query, where a query to a node returns all the shortest paths (i.e., list of nodes in the path) from that node to all other nodes [38]. This is the strongest of all queries. These queries can be implemented by using Traceroute on Internet. In [38], the combinatorial-optimization problem of selecting the smallest subset of nodes for such queries to estimate the network topology is formulated. The work in [39] considers discovery of random graphs using such queries. The bias of using traceroute sampling on power-law graphs is studied in [40], and weighted random walk sampling is considered in [41].
(ii) Distance query, where a query to a node returns all the shortest-path distances (instead of the complete list of nodes) from that node to any other node in the network [39]. These queries are available for instance, in Peer-to-Peer networks through the Ping/Pong protocol. This problem is related to the landmark placement, and the optimization problem of having smallest number of landmarks is known as the metric dimension of the graph [42]. The work in [43] considers reconstruction of tree topologies using shortest-path queries.
(iii) Edge-based queries: There are several types of edge queries such as detection query, which answer whether there is an edge between two selected nodes, or counting query, which returns number of edges in a selected subgraph [44,45], or a cross-additive query, which returns the number of edges crossing between two disjoint sets of vertices [46].

However, all the above queries assume that all the nodes (with labels) are known a priori, and that there are no hidden (unlabeled) nodes in the network. Moreover, most of the above works

consider unweighted graphs, which are not suitable when end-to-end delay (or other weighted) information is available for topology discovery. As previously discussed, the above queries assume extensive information is available from the queried objects, and this may not be feasible in many networks.

Topology discovery using end-to-end delays between a subset of nodes (henceforth, referred to as participating nodes), has been previously studied for tree topologies using unicast traffic in [16,21,47] and multicast traffic [48]. The algorithms are inspired by *phylogenetic* tree algorithms. See [49] for a thorough review. Most of these algorithms are based on a series of local tests known as the *quartet-based distance tests*. Our algorithms are inspired by, and are based on quartet methods. However, these algorithms were previously applied only to tree topologies, and here, we show how algorithms based on similar ideas can provide accurate reconstruction for a much broader class of locally-tree like graphs such as the sparse random graphs. Recent works also incorporate additional information from temporal dynamics [50] or consider causal models for networks [51,52], while our work does not consider these effects.

2 System Model

Notation

For any two functions $f(n), g(n)$, $f(n) = O(g(n))$ if there exists a constant M such that $f(n) \leq Mg(n)$ for all $n \geq n_0$ for some fixed $n_0 \in \mathbb{N}$. Similarly, $f(n) = \Omega(g(n))$ if there exists a constant M' such that $f(n) \geq M'g(n)$ for all $n \geq n_0$ for some fixed $n_0 \in \mathbb{N}$, and $f(n) = \Theta(g(n))$ if $f(n) = \Omega(g(n))$ and $f(n) = O(g(n))$. Also, $f(n) = o(g(n))$ when $f(n)/g(n) \to 0$ and $f(n) = \omega(g(n))$ when $f(n)/g(n) \to \infty$ as $n \to \infty$. We use notation $\tilde{O}(g(n)) = O(g(n)\text{poly}\log n)$. Let $\mathbb{I}[A]$ denote indicator of an event A.

Let G_n denote a random graph with probability measure $\mathbb{P}$. Let $\mathcal{Q}$ be a graph property (such as being connected). We say that the property $\mathcal{Q}$ for a sequence of random graphs $\{G_n\}_{n\in\mathbb{N}}$ holds asymptotically almost surely (a.a.s.) if,

$$\lim_{n\to\infty} \mathbb{P}(G_n \text{ satisfies } \mathcal{Q}) = 1.$$

Equivalently, the property $\mathcal{Q}$ holds for *almost every* (a.e.) graph G_n.

For a graph G, let $\mathcal{C}(l; G)$ denote the set of (generalized) cycles[1] of length less than l in graph G. For a vertex v, let $\text{Deg}(v)$ denote its degree and for an edge e, let $\text{Deg}(e)$ denote the total number of edges connected to either of its endpoints (but not counting the edge e). Let $B_R(v)$ denote the set of nodes within hop distance R from a node v and $\Gamma_R(v)$ is the set of nodes exactly at hop distance R. The definition is extended to an edge, by considering union of sets of the endpoints of edge. Denote the shortest path (with least number of hops) between two nodes i, j as $\text{Path}(i, j; G)$ and the second shortest path as $\text{Path}_2(i, j; G)$. Denote the number of H-subgraphs in G, i.e., the number of subgraphs in G corresponding to H, as $N_{H;G}$.

2.1 Random Graphs

We assume that the unknown network topology is drawn from the ensemble of Erdős-Rényi random graphs [18]. This random graph model is arguably the simplest as well the most well-studied model.

[1] A generalized cycle of length l is a connected graph of l nodes with l edges (i.e., can be a union of a path and a cycle). In this paper, a cycle refers to a generalized cycle unless otherwise mentioned.

Denote the random graph as $G_n \in \mathcal{G}(n, c/n)$, for $c < \infty$, where n is the number of nodes and each edge occurs uniformly with probability c/n. This implies a constant average degree of c for each node, and this regime is also known as the "sparse" regime of random graphs.

It is well known that sparse random graphs exhibit a phase transition with respect to the number of components. When $c > 1$, there is a giant component containing $\Theta(n)$ nodes, while all the other components have size $\Theta(\log n)$ [53, Ch. 11]. This regime is known as the *super-critical* regime. On the other hand, when $c < 1$, there is no giant component and all components have size $\Theta(\log n)$. This regime is known as the *sub-critical* regime.

We consider discovery of a random graph in the super-critical regime ($c > 1$). This is the regime of interest, since most real-world networks are well connected rather than having large number of extremely small components. Moreover, the presence of a giant component ensures that the topology can be discovered even with a small fraction of random participants. This is because the participants will most likely belong the giant component, and can thus exchange messages between each other to discover the unknown topology. We limit ourselves to the topology discovery of the giant component in the random graph, and denote the giant component as G_n, unless otherwise mentioned.

2.2 Participation Model

For the given unknown graph topology $G_n = (W_n, E_n)$ over $W_n = \{1, \ldots, n\}$ nodes, let $V_n \subset W_n$ be the set of participating nodes which exchange messages amongst each other by routing them along the graph. Let $\rho_n := \frac{|V_n|}{n}$ denote the fraction of participating nodes. It is desirable to have small ρ_n and still reconstruct the unknown topology. We assume that the nodes decide to participate uniformly at random. This ensures that information about all parts of the graph can be obtained, thereby making graph reconstruction feasible. We consider the regime, where $|V_n| = n^{1-\epsilon}$, for some $\epsilon > 0$, meaning that extremely small number of nodes participate in discovering the topology.

Let $H_n := W_n \setminus V_n$ be the set of hidden nodes. The hidden nodes only forward the messages without altering them, and do not provide any additional information for topology discovery. The presence of hidden nodes thus needs to be inferred, as part of our goal of discovering the unknown graph topology.

2.3 Delay Model

The messages exchanged between the participating nodes experience delays along the links in the route. The participating nodes measure the end-to-end delays[2] between message transmissions and receptions. We consider the challenging scenario that only this end-to-end delay information is available for topology discovery.

Let m be the number of messages exchanged between each pair of participating nodes $i, j \in V_n$. Denote the m samples of end-to-end delays computed from these messages as

$$\mathbf{D}_{i,j}^m := [D_{i,j}(1), D_{i,j}(2), \ldots, D_{i,j}(m)]^T.$$

We assume that the routes taken by the m messages are fixed, and we discuss the routing model in the subsequent section. On the other hand, these messages experience different delays along

[2]Our algorithms work under any *additive metric* defined on the graph such as link utilization or link loss [16], although the sample complexity, i.e., the number of samples required to accurately estimate the metrics, does indeed depend on the metric under consideration.

each link[3] which are drawn identically and independently (i.i.d) from some distribution, described below.

Let D_e denote the random delay along a link $e \in G_n$ (in either direction). We assume that the delays D_{e_1} and D_{e_2} along any two links $e_1, e_2 \in G_n$ are independent. The delays are additive along any route, i.e., the end-to-end delay along a route $\mathcal{R}(i,j)$ between two participants $i, j \in V_n$ is

$$D_{\mathcal{R}(i,j)} := \sum_{e \in \mathcal{R}(i,j)} D_e. \tag{1}$$

Further, the family of delay distributions are regular and bounded, as in [21].

The delay distributions $\{D_e\}_{e \in E_n}$ and the graph topology G_n are both unknown, and need to be estimated using messages between participating nodes. We exploit the additivity assumption in (1) to obtain efficient topology discovery algorithms.

2.4 Routing Model

The end-to-end delays between the participating nodes thus depends on the routes taken by the messages. We assume that the messages between any two participants are routed along the shortest path with the lowest number of hops. On the other hand, the nodes cannot select the path with the least delay since the delays along the individual links are unknown and are also different for different messages.

We also consider another scenario, where the participants are able to additionally route messages along the second shortest path. This is a reasonable assumption, since in practice, nodes typically maintain information about the shortest path and an alternative path, should the shortest path fail. The nodes can forward messages along the shortest and the second shortest paths with different headers, so that the destinations can distinguish the two messages and compute the end-to-end delays along the two paths. We will show that this additional information vastly improves the accuracy of topology discovery. These two scenarios are formally defined below.

Scenario 1 (Shortest Path Delays): Each pair of participating nodes $i, j \in V_n$ exchange m messages along the shortest path in G_n, where the shortest path[4] is with respect to the number of hops. Denote the vector of m end-to-end delays as $\mathbf{D}_{i,j}^m$.

Scenario 2 (Shortest Path and Second Shortest Path Delays): Each pair of participating nodes $i, j \in V_n$ exchange m messages along the shortest path as well as m messages along the second shortest path. The vector of m samples along the second shortest path is denoted by $\widetilde{\mathbf{D}}_{i,j}^m$.

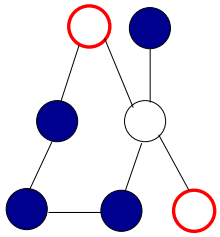
3 Reconstruction Guarantees

3.1 Minimal Representation

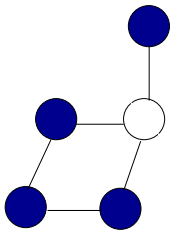
Our goal is to discover the unknown graph topology using the end-to-end delay information between the participating nodes. However, there can be multiple topologies which explain equally well the end-to-end delays between the participants. This inherent ambiguity in topology discovery with hidden nodes has been previously pointed out in the context of latent tree models [54].

[3]The independence assumption implies that we consider unicast traffic rather than multicast traffic considered in many other works, e.g., in [48].

[4]If the shortest path between two nodes is not unique, assume that the node pairs randomly pick one of the paths and use it for all the messages.



(a) A non-minimal graph



(b) Minimal representation

Figure 1: In the above figures, the shaded nodes are participants while the rest are hidden. In the minimal representation of a graph, hidden nodes with degree two and less (the highlighted hidden nodes) are merged with their neighbors. See Procedure 1 for details.

Procedure 1 $\tilde{G}_n := \mathsf{Minimal}(G_n; V_n)$ is the minimal representation of $G_{n'}$ given set of participating nodes V_n.

Input: Graph $G_{n'}$, set of participating nodes V_n, and set of hidden nodes H_n.
Initialize $\tilde{G}_n = G_n$, $n \leftarrow n'$.
while $\exists h \in \tilde{G}_n \cap H_n$ with $\mathrm{Deg}(h) \leq 2$ **do**
  Remove h from $\tilde{G}_n$ if $\mathrm{Deg}(h) \leq 1$.
  Contract all h with $\mathrm{Deg}(h) = 2$ in $\tilde{G}_n$.
  Decrement n accordingly.
end while

There is an equivalence class of topologies with different sets of hidden nodes which generate the same end-to-end delay distributions between the participating nodes. We refer to the topology with the least number of hidden nodes in this equivalence class as the *minimal representation*. Such a minimal representation does not have redundant hidden nodes. For example, in Fig.1, the graph and its minimal representation are shown. In Procedure 1, we characterize the relationship between a graph and its minimal representation, given a set of participants. The minimal representation is obtained by iteratively removing redundant hidden nodes (degree two and less) from the graph, i.e., in the first iteration, redundant hidden nodes are removed and the resulting graph is again inspected for the presence of hidden nodes. For example, in Fig.1, the highlighted hidden nodes are redundant and are thus merged with their neighbors to obtain the minimal representation.

Any algorithm can only reconstruct the unknown topology up to its minimal representation using only end-to-end delay information between the participating nodes. In sparse random graphs, only a small (but a linear) number of nodes are removed in the minimal representation, and this number decreases with the average degree c. It thus suffices to reconstruct the minimal representation of the original topology, and our goal is to accomplish it using small fraction of participants. We assume that the delay distributions on the edges of the minimal representation $\{D_e\}_{e\in\widetilde{G}_n}$ have bounded variances $\{l(e)\}_{e\in\widetilde{G}_n}$ satisfying

$$0 < f \leq l(e) \leq g < \infty, \quad \forall\, e \in \widetilde{G}_n. \tag{2}$$

3.2 Performance Measures

We now define performance measures for topology discovery algorithms. It is desirable to have an algorithm which outputs a graph structure which is close to the original graph structure. However, the reconstructed graph cannot be directly compared with the original graph since the hidden nodes introduced in the reconstructed graph are unlabeled and may correspond to different hidden nodes in the original graph. To this end, we require the notion of edit distance defined below.

Definition 1 (Edit Distance) *Let F, G be two graphs[5] with adjacency matrices $\mathbf{A}_F, \mathbf{A}_G$, and let V be the set of labeled vertices in both the graphs (with identical labels). Then the edit distance between F, G is defined as*

$$\Delta(F, G; V) := \min_{\pi} ||\mathbf{A}_F - \pi(\mathbf{A}_G)||_1,$$

where π is any permutation on the unlabeled nodes while keeping the labeled nodes fixed.

In other words, the edit distance is the minimum number of entries that are different in $\mathbf{A}_F$ and in any permutation of $\mathbf{A}_G$ over the unlabeled nodes. In our context, the labeled nodes correspond to the participating nodes while the unlabeled nodes correspond to hidden nodes.

Our goal is to output a graph with small edit distance with respect to the minimal representation of the original graph. Ideally, we would like the edit distance to decay as we obtain more delay samples and this is the notion of consistency.

Definition 2 (Consistency) *Denote $\widehat{G}_n(\{\mathbf{D}^m_{i,j}\}_{i,j\in V_n})$ as the estimated graph using m delay samples between the participating nodes V_n. A graph estimator $\widehat{G}_n(\{\mathbf{D}^m_{i,j}\}_{i,j\in V_n})$ is* structurally consistent *if it asymptotically recovers the minimal representation of the unknown topology, i.e.,*

$$\lim_{m\to\infty} \mathbb{P}[\Delta(\widehat{G}(\{\mathbf{D}^m_{i,j}\}_{i,j\in V_n}), \tilde{G}_n; V_n) > 0] = 0. \tag{3}$$

The above definition assumes that the network size n is fixed while the number of samples m goes to infinity. A more challenging setting where both the network size and the number of samples grow is known as the setting of *high-dimensional inference* [17]. In this setting, we are interested in estimating large network structures using a small number of delay samples. We will consider this setting for topology discovery in this paper. Indeed in practice, we have large network structures but can obtain only few end-to-end delay samples with respect to the size of the network. This is formalized using the notion of sample complexity defined below for our setting.

Definition 3 (Sample Complexity) *If the number of samples is $m = \Omega(f(n))$, for some function f, such that the estimator $\widehat{G}_n(\{\mathbf{D}^m_{i,j}\}_{i,j\in V_n})$ satisfies*

$$\lim_{\substack{m,n\to\infty \\ m=\Omega(f(n))}} \mathbb{P}[\Delta(\widehat{G}(\{\mathbf{D}^m_{i,j}\}_{i,j\in V_n}), \tilde{G}_n; V_n) = O(g(n))] = 0,$$

for some function $g(n)$, then the estimator $\widehat{G}_n$ is said to have sample complexity of $\Omega(f(n))$ for achieving an edit distance of $O(g(n))$.

Thus, our goal is to discover topology in high-dimensional regime, and design a graph estimator that requires a small number of delay samples, and output a graph with a small edit distance.

[5] We consider inexact graph matching where the unlabeled nodes can be unmatched. This is done by adding required number of isolated unlabeled nodes in the other graph, and considering the modified adjacency matrices [55].

4 Preliminaries

We now discuss some simple concepts which will be incorporated into our topology discovery algorithms.

4.1 Delay Variance Estimation

In our setting, topology discovery is based on the end-to-end delays between the participating nodes. Recall that in Section 2.3, we assume general delay distributions on the edges with bounded variances. Our topology discovery algorithms will be based solely on the estimated variances using the end-to-end delay samples.

We use the standard unbiased estimator for variances [56].

$$\widehat{l}^m(i,j) := \frac{1}{m-1}\sum_{k=1}^{m}(D_{i,j}(k) - \bar{D}^m_{i,j})^2, \tag{4}$$

where $\bar{D}^m_{i,j}$ is the sample mean delay

$$\bar{D}^m_{i,j} := \frac{1}{m}\sum_{k=1}^{m} D_{i,j}(k). \tag{5}$$

Note that we do not use an estimator specifically tailored for a parametric delay distribution, and hence, the above estimator yields unbiased estimates for any delay distribution.

Our proposed algorithms for topology discovery require only the estimated delay variances $\{\widehat{l}(i,j)\}_{i,j\in V}$ as inputs. Indeed, more information is available in the delay samples $\mathbf{D}^m$. For instance, in [21], the higher-order moments of the delay distribution are estimated using the delay samples and this provides an estimate for the delay distribution. However, we see that for our goal of topology discovery, the estimated end-to-end delay variances suffice and yield good performance.

Recall that $\{l(i,j)\}_{i,j\in V}$ denotes the true end-to-end delay variances and that from (1), the variances are additive along any path in the graph. We will henceforth refer to the variances as "distances" between the nodes and the estimated variances as "estimated distances". This abstraction also implies that our algorithms will work under input of estimates of any additive metrics.

4.2 Quartet Tests

We first recap the so-called quartet tests, which are building blocks of many algorithms for discovering phylogenetic-tree topologies with hidden nodes [54, 57–59]. The definition of a quartet is given below. See Fig.2.

Definition 4 (Quartet or Four-Point Condition) *The pairwise distances $\{l(i,j)\}_{i,j\in\{a,b,u,v\}}$ for the configuration in Fig.2 satisfy*

$$l(a,u) + l(b,v) = l(b,u) + l(a,v), \tag{6}$$

and the configuration is denoted by $Q(ab|uv)$.

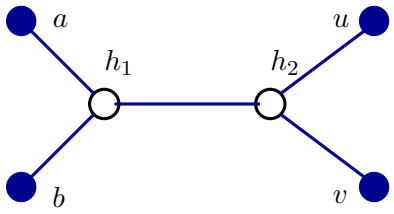


Figure 2: Quartet $Q(ab|uv)$. See (6) and (8).

In the literature on tree reconstruction, instead of (6), an inequality test is usually employed since it is more robust, given by,

$$l(a,b) + l(u,v) < \min(l(a,u) + l(b,v), l(b,u) + l(a,v)). \tag{7}$$

However, we use the equality test in (6), since it is also useful in detecting cycles present in random graphs.

In practice, we only have access to distance estimates and we relax the equality constraint in (6) to a threshold test, and this is known as the quartet test. Thus, the quartet test is local test between tuples of four nodes. For the quartet $Q = (ab|uv)$, let e denote the middle edge of the quartet[6], i.e., the edge which joins a vertex on the shortest path between a and b to a vertex on the shortest path between u and v (Note that the edge can have zero length if the hidden nodes connecting a, b and u, v are the same.). The estimated length of the middle edge (h_1, h_2) between hidden nodes h_1 and h_2 is given by

$$2\widehat{l}(h_1,h_2) = \widehat{l}(a,u) + \widehat{l}(b,v) - \widehat{l}(a,b) - \widehat{l}(u,v). \tag{8}$$

Similarly, all other edge lengths of the quartet can be calculated through the set of linear equations which are based on the fact that the end-to-end lengths in a quartet are the sum of edge lengths along the respective paths.

Many phylogenetic-tree reconstruction algorithms proceed by iteratively merging quartets to obtain a tree topology. See [60] for details. We employ the quartet test for random graph discovery but it additionally incorporates the presence of cycles. Moreover, we introduce modifications under scenario 2, as outlined in Section 5.2, where second shortest path distances are available in addition to the shortest path distances between the participating nodes.

5 Proposed Algorithms

5.1 Scenario 1

We propose the algorithm RGD1 for discovering random graphs under scenario 1, as outlined in Section 2.4, where only shortest path distance estimates are available between the participating nodes. The idea behind RGD1 is similar to the classical phylogenetic-tree reconstruction algorithms based on quartet tests [54, 58]. However, the effect of cycles on such tests needs to analyzed, and is carried out in Section 6.1. The algorithm is summarized in Algorithm 2.

[6]Such a middle edge always exists, by allowing for zero length edges, and such trivial edges are contracted later in the algorithm.

The algorithm recursively runs the quartet tests over the set of participating nodes. The algorithm limits to testing only "short quartets" between nearby participating nodes. Intuitively, this is done to avoid testing quartets on short cycles, since in such scenarios, the quartet tests may fail to reconstruct the graph accurately. Since the random graphs are locally-tree like and contain a small number of short cycles, limiting to short quartets enables us to avoid most of the cycles. The idea of short quartets has been used before (e.g. in [58]) but for a different goal of obtaining low sample complexity algorithm for phylogenetic-tree reconstruction. We carry out a detailed analysis on the effect of cycles on quartet tests in Section 6.1.

In algorithm RGD1, we consider short quartets, where all the estimated distances between the quartet end points are at most $Rg + \tau$, where g is the upper bound on the (exact) edge lengths in the original graph, as assumed in (2). Thus, $R' := Rg/f$ is the maximum number of hops between the end points of a short quartet, where f is the lower bound on the edge lengths. We refer to R' as the *diameter of the quartet*. This needs to be chosen carefully to balance the following two events: encountering short cycles and ensuring that most hidden edges (with at least one hidden end point) are part of short quartets. The parameter τ is chosen to relax the bound, since we have distance estimates, computed using samples, rather than exact distances between the participating nodes. The short quartets are listed in arbitrary order in $\mathcal{Q}$.

The algorithm attempts to merge the quartets in $\mathcal{Q}$, one at a time, with the previously constructed graph $\widehat{G}_n$ using procedure QuartetMerge. There are different possibilities during this process. The quartet under consideration, say $Q(ab|uv)$, may be already satisfied in $\widehat{G}_n$: nothing needs to be done in such a scenario; or the quartet may be merged without creating new cycles. This is carried out using procedure TreeMerge. Alternatively, if a cycle needs to be created in $\widehat{G}_n$ to merge $Q(ab|uv)$, additional testing needs to be carried out. Firstly, if it is a short cycle (of length less than $2Rg+\tau$), then the algorithm cannot be guaranteed to merge $Q(ab|uv)$ accurately and it is listed as a bad quartet. Secondly, if it is not a short cycle, the algorithm needs to infer the joining points between the existing paths in $\widehat{G}_n$ and the new path to be created. This is carried out using procedure CycleMerge and entails the presence of "witnesses" $\mathcal{W} \subset \mathcal{Q}$, which are (remaining) short quartets whose nodes are within distance $2R$ from a, b, u, v. The algorithm attempts to merge the quartets in $\mathcal{W}$ without creating new cycles in $\widehat{G}_n$, and then attempts to merge $Q(ab|uv)$ by using existing hidden nodes in the paths to create a new (long) cycle and checking if it conflicts with the distances on quartets in $\mathcal{W}$. There is a tolerance of ϵ' for checking distance conflicts. In the end, any edge smaller than a threshold ϵ are contracted, for some chosen constant $\epsilon < f$, where f is the lower bound on the edge lengths of the original graph.

The quartets that fail to be merged using the above procedure are listed as bad quartets. These set of quartets cannot be guaranteed to be merged accurately. Any post-processing heuristic can be used to attempt the merging of these bad quartets. Our analysis accounts for these bad quartets towards contributing to the edit distance between the reconstructed graph and the minimal representation of the original graph. The above algorithm is similar in spirit to quartet merging algorithm proposed in [58], but with the crucial addition of CycleMerge procedure to handle the presence of cycles.

5.2 Scenario 2

We now consider scenario 2, as outlined in Section 2.4, where second shortest path distance estimates are available in addition to shortest path distance estimates between the participating nodes. We propose RGD2 algorithm for this case, which is summarized in Algorithm 3.

Algorithm 2 RGD1($\{\widehat{l}(i,j)\}_{i,j\in V_n}; R, g, \tau, \epsilon, \epsilon'$) for Topology Discovery Using Shortest-Path Distance Estimates.

Input: Distance estimates between the participating nodes $\{\widehat{l}(i,j)\}_{i,j\in V_n}$, upper bound g on exact edge lengths and parameters $R, \tau, \epsilon, \epsilon' > 0$.
Initialize list of short quartets: $\mathcal{Q} = \{Q(ab|uv) : \max_{i,j\in\{a,b,u,v\}} \widehat{l}(i,j) < Rg + \tau\}$, list of bad quartets $\mathcal{Q}_{\text{bad}} = \emptyset$ and reconstructed graph $\widehat{G}_n = (V_n, \emptyset)$.
while $\mathcal{Q} \neq \emptyset$ **do**
  $Q(ab|uv) \leftarrow \text{Pop}(\mathcal{Q})$.
  $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \{Q(ab|uv)\}$.
  $[\widehat{G}_n, \text{Fail}] \leftarrow \textsf{QuartetMerge}(\widehat{G}_n, Q(ab|uv), \mathcal{Q}, \{\widehat{l}(i,j)\}_{i,j\in V_n}; \epsilon, \epsilon')$.
  if Fail **then**
    $\mathcal{Q}_{\text{bad}} \leftarrow \mathcal{Q}_{\text{bad}} \cup \{Q(ab|uv)\}$.
  end if
end while
Use any heuristic to merge bad quartets in $\mathcal{Q}_{\text{bad}}$.

The algorithm RGD2 is an extension of RGD1, where we use the second shortest distances in the quartet tests, in addition to the shortest distances. For each tuple of participating nodes $a, b, u, v \in V_n$, the quartet test in (6) is carried out for all possible combinations of shortest and second shortest distances; only short quartets are retained, where all the distances used for quartet test are less than the specified threshold (which is the same as in RGD1). If the same quartet is formed using different combinations of shortest and second shortest distances, only the quartet with the shorter middle edge, computed using (8), is retained. We clarify the reason behind this rule and give examples on when this can occur in Section 6.1. As before, all these quartets are merged with previously constructed graph using procedure QuartetMerge, but with a minor difference that the path lengths need to be checked since there may be multiple paths between participating nodes with different lengths. The performance analysis for RGD2 is carried out in Section 6.3.

6 Analysis Under Exact Distances

We now undertake performance analysis for the proposed topology discovery algorithms RGD1 and RGD2. In this section, for simplicity, we first analyze the performance assuming that exact distances between the participating nodes are input to the algorithms. Analysis when distance estimates are input to the algorithms is considered in Section 7.

6.1 Effect of Cycles on Quartet Tests

We now analyze the effect of cycles on quartet tests. Recall that the quartet test is the inequality test in (6), and if this inequality test is satisfied, internal edge lengths of the quartet are computed, and they are added to the output using procedure QuartetMerge. The quartet test in (6) is based on the assumption that the shortest paths between the four nodes $\{a, b, u, v\}$ in the quartet are along the paths on the quartet.

Thus, the outcome of the quartet test is incorrect only when some shortest path between $\{a, b, u, v\}$ is outside the quartet. We refer to such quartets as "bad quartets". There are two

Algorithm 3 RGD2($\{\widehat{l}(i,j), \widehat{l}_2(i,j)\}_{i,j\in V_n}; R, g, \tau, \epsilon, \epsilon'$) for Topology Discovery Using Shortest-Path and Second Shortest-Path Distance Estimates.

Input: Shortest-path and second shortest-path distance estimates $\{\widehat{l}(i,j), \widehat{l}_2(i,j)\}_{i,j\in V_n}$ upper bound g on exact edge lengths and parameters $R, \tau, \epsilon, \epsilon' > 0$.
Initialize list of short quartets: $\mathcal{Q} = \{Q(ab|uv) : \max_{i,j\in\{a,b,u,v\}} \widehat{l}(i,j) < Rg + \tau\}$, list of bad quartets $\mathcal{Q}_{\text{bad}} = \emptyset$ and reconstructed graph $\widehat{G}_n = (V_n, \emptyset)$.
while $\mathcal{Q} \neq \emptyset$ **do**
  $Q(ab|uv) \leftarrow \text{Pop}(\mathcal{Q})$.
  $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \{Q(ab|uv)\}$.
  $[\widehat{G}_n, \text{Fail}] \leftarrow \text{QuartetMerge}(\widehat{G}_n, Q(ab|uv), \mathcal{Q}, \{\widehat{l}(i,j), \widehat{l}_2(i,j)\}_{i,j\in V_n}; \epsilon, \epsilon')$.
  if Fail **then**
    $\mathcal{Q}_{\text{bad}} \leftarrow \mathcal{Q}_{\text{bad}} \cup \{Q(ab|uv)\}$.
  end if
end while
Use any heuristic to merge bad quartets in $\mathcal{Q}_{\text{bad}}$.

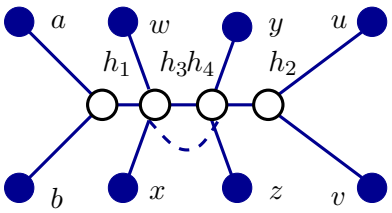


Figure 3: An example on the use of witnesses for creating new paths in the existing graph $\widehat{G}$ in CycleMerge Procedure. In order to create a new path between h_1 and h_2 (shown using dotted lines according to length specified by quartet $Q(ab|uv)$), the quartet $Q(wx|yz)$ is used as a witness to verify if h_3 and h_4 are the joining points of the existing path in $\widehat{G}$ with the new path.

possible outcomes for bad quartets (a) the procedure QuartetMerge detects inconsistencies in the set of linear equations, used to compute the internal distances in the quartet, and does not merge the quartet, or (b) the procedure QuartetMerge does not detect inconsistencies, and thus merges a fake quartet with wrong internal edge lengths. Both these outcomes result in reconstruction error.

The examples of both the cases are given in Fig.5. Note that the set of linear equations used by the procedure QuartetMerge for computing the internal edge-lengths in the quartet consist of 5 variables and 6 equations (corresponding to the 6 known edge-lengths between the quartet end-points). Additionally, there is an equality constraint that $l(a,u)+l(b,v) = l(a,v)+l(b,u)$. The case in Fig.5a does not satisfy this equality constraint[7], since the cycle is in the middle of the quartet, and thus the procedure QuartetMerge does not merge this quartet. On the other hand, for the case in Fig.5b, the equality constraint is satisfied, since the cycle is on the same side of the quartet, and in this case, the procedure QuartetMerge merges the quartet, but with wrong edge lengths, as shown in Fig.5c.

Thus, bad quartets lead to reconstruction error. The number of bad quartets can be bounded

[7]There exist pathological cases of equal distances where configurations of the form in Fig.5a will satisfy equality constraint. Such scenarios do not occur in a.e. random graph.

Procedure 4 $\mathsf{QuartetMerge}(\widehat{G}_n, Q(ab|uv)), \mathcal{Q}, \{\widehat{l}(i,j), [\widehat{l}_2(i,j)]\}_{i,j\in V_n}; \epsilon, \epsilon')$ for merging a new quartet $Q(ab|uv)$ with current structure $\widehat{G}_n$.

Input: Current graph $\widehat{G}$, candidate quartet $Q(ab|uv)$, remaining short quartets $\mathcal{Q}$, shortest distance estimates between the participating nodes $\{\widehat{l}(i,j)\}_{i,j\in V_n}$ and optionally second shortest distances $\{\widehat{l}_2(i,j)\}_{i,j\in V_n}$, threshold ϵ for contracting short edges and tolerance ϵ' for comparing path lengths.

if For each $i,j \in \{a,b,u,v\}$, $|\widehat{l}(i,j;\widehat{G}) - \widehat{l}(i,j;Q)| < \epsilon'$ (All paths already present in $\widehat{G}$) **then**
  Fail $\leftarrow$ False.
else if For each $i,j \in \{a,b,u,v\}$, either $|\widehat{l}(i,j;\widehat{G}) - \widehat{l}(i,j;Q)| < \epsilon'$ or $\widehat{l}(i,j;\widehat{G}) = \infty$ (Either the paths agree or the path does not exist in $\widehat{G}$) **then**
  $\widehat{G}_n \leftarrow \mathsf{TreeMerge}(\widehat{G}_n, Q(ab|uv), \epsilon')$, Fail $\leftarrow$ False. (Merge quartet without creating cycles).
else if $\exists\, i,j \in \{a,b,u,v\}$ such that $\widehat{l}(i,j;\widehat{G}) + \widehat{l}(i,j;Q) < 2Rg + \tau$ (A merge would create a short cycle) **then**
  if Second shortest distances $\{\widehat{l}_2(i,j)\}_{i,j\in V_n}$, are available **then**
    Use second shortest distances between a,b,u,v to infer the join points in $\widehat{G}$. If the points are consistently found, add quartet $Q(ab|uv)$ to $\widehat{G}$ and output Fail $\leftarrow$ False. Else output Fail $\leftarrow$ True. (Report failure due to inconsistent distances).
  else
    Fail $\leftarrow$ True. (Report failure due to presence of a short cycle).
  end if
else
  $[\widehat{G}_n, \text{Fail}] \leftarrow \mathsf{CycleMerge}(\widehat{G}_n, Q(ab|uv), \mathcal{Q}, \{\widehat{l}(i,j), [\widehat{l}_2(i,j)]\}_{i,j\in V_n}, [\{\widehat{l}_2(i,j)\}_{i,j\in V_n}]; \epsilon')$. (Attempt to merge quartet by creating a new long cycle and querying witnesses, if no witnesses are present, create a new path, else if witnesses are contradictory output fail).
end if
Contract any edge (with at least one hidden end point) with length $< \epsilon$.

as follows: in a bad quartet, the middle edge of the quartet is part of a (generalized) cycle of length less than $2R'$, where $R' := Rg/f$ is the maximum number of hops between the endpoints of a short quartet, as discussed in Section 5.1. In addition, the bad quartets also affect the merging of quartets using CycleMerge procedure when they are called upon to serve as witnesses. Thus, we also need to consider quartets which are part of slightly longer cycles. See Appendix B for details. The number of such bad quartets can be bounded for random graphs leading to reconstruction guarantees for RGD1 algorithm.

For the RGD2 algorithm where second shortest path distances are additionally available, bad quartets do not adversely affect performance. We argue that a quartet is correctly recognized as long as the paths on the quartet correspond to either the shortest or the second shortest paths (between the quartet endpoints). In such a scenario, some combination of shortest and second shortest path distances exists which accurately reconstructs the quartet and the RGD2 algorithm finds all such combinations. Moreover, fake quartets are detected since they produce a longer middle edge than the true quartet. This is because the cycle shortens the distance between end points on its side (in Fig.5b, this corresponds to $\{a,b\}$ and note that the middle edge in Fig.5c is longer than the true edge length).

Thus, a quartet is correctly reconstructed under RGD2 when the paths on the quartet consist of

Procedure 5 TreeMerge$(\widehat{G}_n, Q(ab|uv)); \epsilon')$ for merging a new quartet with current structure $\widehat{G}_n$ without creating cycles.

Input: Current graph $\widehat{G}$, candidate quartet $Q(ab|uv)$ with hidden nodes h_1, h_2 (See Fig.2), and tolerance ϵ' for comparing path lengths.

if There exists hidden node in $\widehat{G}$ such that $|\widehat{l}(i,h;\widehat{G}) - \widehat{l}(i,h_1;Q)| < \epsilon'$ for $i = a, b$ **then**
Assign h as h_1.
end if

if There exists hidden node in $\widehat{G}$ such that $|\widehat{l}(i,h;\widehat{G}) - \widehat{l}(i,h_2;Q)| < \epsilon'$ for $i = u, v$ **then**
Assign h as h_2.
end if

Connect paths in $\widehat{G}$ according to Q which are missing as follows:

If both h_1 and h_2 are assigned in $\widehat{G}$, connect h_1 and h_2 in $\widehat{G}$ and assign length $\widehat{l}(h_1, h_2; Q)$ if they are not already connected in $\widehat{G}$.

If say h_1 is assigned and the path between a and h_1 exists in $\widehat{G}$ but not between b and h_1, let $l \leftarrow \min_w 0.5(\widehat{l}(w,b) + \widehat{l}(h_1,b;Q) - \widehat{l}(h_1,w;\widehat{G})$ over all $w \in \widehat{G}$ such that $\widehat{l}(w,b) < Rg + \tau$. If $l < \widehat{l}(h_1,b;Q)$, split path $\widehat{l}(h_1,w;\widehat{G})$, add new hidden node h_3 such that $\widehat{l}(h_3,w;\widehat{G}) = \widehat{l}(b,w) - l$ and attach b to h_3 with length l; otherwise create a new path between h_1 and b. Similarly split the other paths if present or create new paths.

If the paths exist but not the hidden nodes, split the paths and add hidden nodes according to the lengths in Q. Otherwise, also add new paths to $\widehat{G}$.

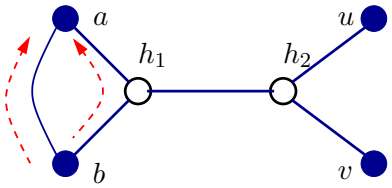


Figure 4: A bad quartet: $l(a,b) < l(a,h_1) + l(b,h_1)$. Since the maximum number of hops between $\{a,b,u,v\}$ is $R' := Rg/f$, and one of the shortest paths is not along the quartet, the middle edge (h_1,h_2) is part of a generalized cycle of (hop) length less than $2R'$. Such quartets are detected by the RGD1 algorithm.

shortest or second shortest paths. We finally use the locally tree-like property of random graphs to establish that this occurs in almost every graph if the quartet diameter R' is small enough. Thus, we obtain stronger reconstruction guarantees for RGD2 algorithm.

6.2 Analysis of RGD1

We now provide edit distance guarantees for RGD1 under appropriate choice of maximum quartet diameter $R' := Rg/f$. We analyze the edit distance by counting the number of hidden edges (with at least one hidden end point) which are not recovered correctly under RGD1. A hidden edge is not recovered when one of the following two events occur: (a) it is not part of a short quartet (b) it is part of a bad short quartet. A large value of the quartet diameter R' decreases the likelihood of event (a), while it increases the likelihood of event (b), i.e., we are likely to encounter more cycles as R' is increased. For a fixed value of R', we analyze the likelihood of these two events and obtain

Procedure 6 $\mathsf{CycleMerge}(\widehat{G}_n, Q(ab|uv)), \mathcal{Q}, \{\widehat{l}(i,j)\}_{i,j\in V_n}; \epsilon')$ for merging a new quartet with current structure $\widehat{G}_n$ by creating cycles.

Input: Current graph $\widehat{G}$, candidate quartet $Q(ab|uv)$ with hidden nodes h_1, h_2 (See Fig.2), remaining short quartets $\mathcal{Q}$, shortest distance estimates $\{\widehat{l}(i,j)\}_{i,j\in V_n}$ and optionally second shortest distance estimates $\{\widehat{l}_2(i,j)\}_{i,j\in V_n}$ and tolerance ϵ' for comparing path lengths.
$\mathcal{W} \leftarrow \{Q(ij|kl) : Q(ij|kl) \in \mathcal{Q}, \cup_{x,y\in\{i,j,k,l,a,b,u,v\}} \widehat{l}(x,y) < 2Rg + \tau\}$. (Find potential witnesses by using short quartets "near" to a, b, u, v. Also use second shortest distances if available and they are less than $2Rg + \tau$).
for Each $Q(wx|yz) \in \mathcal{W}$ **do**
  if For $i, j \in \{w, x, y, z\}$, either $|\widehat{l}(i,j;\widehat{G}) - \widehat{l}(i,j;Q)| < \epsilon'$ or $\widehat{l}(i,j;\widehat{G}) = \infty$ (Either the paths agree in the two graphs or the path does not exist in $\widehat{G}$) **then**
    $\widehat{G}_n \leftarrow \mathsf{TreeMerge}(\widehat{G}_n, Q(wx|yz), \epsilon')$, Fail $\leftarrow$ False. (Merge all quartets in $\mathcal{W}$ which do not create cycles).
  end if
end for
To create new paths in $\widehat{G}$ according to $Q(ab|uv)$, consider all hidden nodes on paths in $\widehat{G}$ as candidates for positions where the paths split. Query the quartet corresponding to these hidden nodes for verification. (See Fig.3 for an example).
If the witnesses are absent or contradictory, output Fail $\leftarrow$ True. Else, add the new path to $\widehat{G}$ and output Fail $\leftarrow$ False.

the bound on edit distance stated below.

Assume that the algorithm RGD1 chooses parameter R as

$$R_{\min} \le R \le R_{\max}, \tag{9}$$

where

$$R_{\min} := 2\frac{\log \frac{9\log n}{(\sqrt{c}-1)^2}}{\log 3}, \quad R_{\max} := \frac{6\log n}{5\log c}. \tag{10}$$

Let the fraction of participating nodes be $\rho_n = n^{-\beta}$, such that

$$\rho_n c^{\frac{R-R_{\min}}{2}} = \omega(1), \tag{11}$$

implying that $\gamma > 2\beta$, where

$$\gamma := \frac{\log c}{\log n}(R - R_{\min}). \tag{12}$$

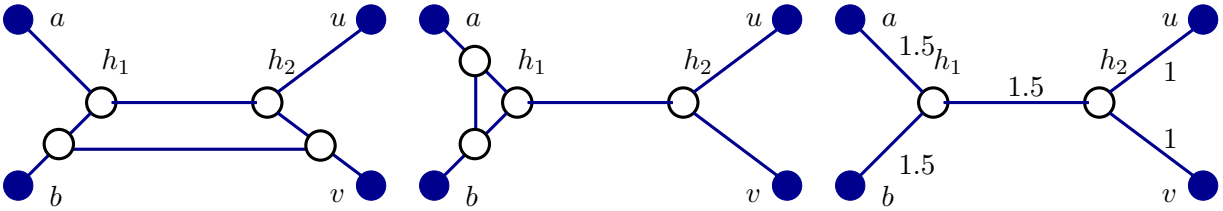
Similarly, define μ as

$$\mu := R\frac{\log \frac{c}{\xi(c)}}{\log n}, \tag{13}$$

where $\xi(c)$ is a function that depends on the average degree c of the original Erdős-Rényi random graph, and is given by

$$\xi(c) := 1 - e^{-c} - ce^{-c} - 0.5c^2e^{-c}. \tag{14}$$

Recall that f and g are the bounds on edge lengths according to (2). We have the following result.



(a) Bad Quartet Where Test Fails. (b) Bad Quartet Where Test Succeeds. (c) Reconstructed Quartet for (b).

Figure 5: Two possible outcomes for bad quartets in (a) and (b). Assume all unit-length edges. In (a), the procedure QuartetMerge fails and quartet is not declared, while in (b), it succeeds but leads to wrong edge estimates, as shown in (c).

Theorem 1 (Edit Distance Under RGD1) *The algorithm* RGD1 *recovers the minimal representation* $\widetilde{G}_n$ *of the giant component of a.e. graph* $G_n \sim \mathcal{G}(n, c/n)$ *with edit distance*

$$\Delta(\widehat{G}_n, \widetilde{G}_n; V_n) = \tilde{O}(n^{5\mu g/f - 4\beta}). \tag{15}$$

Remarks:
(i) Thus, an edit-distance guarantee can be provided under RGD1 when the parameter R is chosen according to the constraints mentioned above. A sufficient condition to achieve a sub-linear edit distance above under homogeneous edge lengths ($f = g$) is when

$$10\beta(1+\delta)\frac{\log \frac{c}{\xi(c)}}{\log c} - 4\beta < 1, \tag{16}$$

for some constant $\delta > 0$. When $c \to \infty$, we have $\xi(c) \to 1$ and in this regime, we have that $\beta < 1/6$. In other words, approximately $n^{\frac{5}{6}}$ nodes need to participate to achieve a sub-linear edit distance under RGD1.
(ii) When the ratio of the bounds on the edge lengths g/f is small (i.e., the edge lengths are nearly homogeneous), the edit-distance guarantee in (15) improves, for a fixed ρ. This is because we can control the hop lengths of the selected quartets more effectively in this case.
(iii) The dominant event leading to the edit-distance bound in (15) is the presence of bad quartets due to short cycles in the random graph. In subsequent section, we show that RGD2 algorithm effectively handles this event using the second shortest path distances.
Proof Ideas:

The proof is based on the error events that can cause the quartet tests to fail. The first error event is that an edge which does not occur as a middle edge of a short quartet, meaning that there are not enough participating nodes within distance $R/2$ from it. The second error event is that an edge occurs as a middle edge of a bad quartet, meaning that it is close to a short cycle or it has bad quartets as witnesses. We analyze the probability of these events and the resulting edit distance due to these events.

6.3 Analysis of RGD2

We now provide edit distance guarantees for RGD2 algorithm. The analysis is on the lines of the previous section, but we instead analyze the presence of overlapping cycles, as noted in Section 6.1.

There are no overlapping short cycles in a random graph, and thus, we can provide a much stronger reconstruction guarantee for the RGD2 algorithm, compared to the RGD1 algorithm. We have the following result.

Theorem 2 (Edit Distance Under RGD2) *Under the assumptions of Theorem 1, the algorithm* RGD2 *recovers the minimal representation $\widetilde{G}_n$ of the giant component of a.e. graph $G_n \sim \mathcal{G}(n, c/n)$ with edit distance*

$$\Delta(\widehat{G}_n, \widetilde{G}_n; V_n) = \tilde{O}(n^{8\mu g/f - 4\beta - 1}). \tag{17}$$

The above result immediately implies that consistent recovery of the minimal representation is possible when there are enough number of participating nodes. We state the result formally below.

Corollary 1 (Consistency Under RGD2) *The algorithm* RGD2 *consistently recovers the minimal representation $\tilde{G}_n$ of the giant component of a.e. graph $G_n \sim \mathcal{G}(n, c/n)$, when the parameter R and the fraction of participating nodes ρ satisfy*

$$\left(\frac{c}{\xi(c)}\right)^{\frac{8Rg}{f}} \rho^4 = o(n), \quad c^{\frac{R-R_{\min}}{2}} \rho = \omega(1),$$

or equivalently

$$\frac{8\mu g}{f} - 4\beta < 1, \quad \gamma > 2\beta.$$

Remarks:
(i) From the above constraints, we see that consistent topology recovery is feasible. Thus, for homogeneous edge lengths ($f = g$), as $c \to \infty$ and the number of participants is more than $n^{11/12}$, RGD2 consistently recovers the topology. Thus, a sub-linear number of participants suffice to recover the minimal representation consistently.
(ii) Thus, the availability of second shortest distances makes consistent topology discovery possible with a sub-linear number of participating nodes, while consistent recovery is not tractable under RGD1 using only shortest-path distances between a sub-linear number of participants.
Proof Ideas:

The proof is on similar lines as in Theorem 1, but with modified error events that cause the quartet tests to fail. As before, the first error event is that an edge which does not occur as a middle edge of a short quartet. The second error event is now that an edge is close to two overlapping short cycles instead of being close to a single short cycle. This event does not occur in random graphs for sufficiently short lengths, and thus, we see a drastic improvement in edit distance.

7 Analysis Under Samples

We have so far analyzed the performance of RGD1 and RGD2 algorithms when exact distances (i.e., delay variances) are input to the algorithm. We now analyze the scenario when instead only delay samples are available and estimated variances are input to the algorithm.

We show that the proposed algorithms have low sample complexity, meaning they require slow scaling of number of samples compared to the network size to achieved guaranteed performance. The result is given below.

Theorem 3 (Sample Complexity) *The edit distance guarantees under* RGD1 *and* RGD2 *algorithms, as stated in Theorem 1 and Theorem 2, are achieved under input of estimated delay variances, if the number of delay samples satisfies*

$$m = \Omega(\text{poly}(\log n)). \tag{18}$$

Thus, the sample complexity of RGD1 and RGD2 algorithms is $\text{poly}(\log n)$. In other words, the size of the network n can grow much faster than the number of delay samples m, and we can still obtain good estimates of the network. This implies with $m = \Omega(\text{poly}(\log n))$ samples, we can consistently discover the topology under RGD2 algorithm, given sufficient fraction of participating nodes.

Proof Ideas:

The proof follows from Azuma-Hoeffding inequality for concentration of individual variance estimates, as in [21, Proposition 1], and then consider the union bound over various events.

8 Converse Results & Discussion

8.1 Fraction of Participating Nodes

We have so far provided edit distance guarantees for the proposed topology discovery algorithms. In this section, we provide a lower bound on the fraction of participating nodes required for any algorithm to recover the original graph up to a certain edit distance guarantee.

We can obtain a meaningful lower bound only when the specified edit distance is lower than the edit distance between a given graph and an independent realizations of the random graph. Otherwise, the edit distance guarantee could be realized by a random construction of the output graph. To this end, we first prove a lower bound on the edit distance between any fixed graph and an independent realization of the random graph.

Let $\mathcal{D}(G;\delta)$ denote the set of all graphs which have edit distance of at most δ from G

$$\mathcal{D}(G;\delta) := \{F : \Delta(F, G; \emptyset) < \delta\}. \tag{19}$$

Lemma 1 (Lower Bound on Edit Distance) *Almost every random graph $G_n \sim \mathcal{G}(n, c/n)$ has an edit distance at least $(0.5c - 1)n$ from any given graph F_n.*

Proof: First, we have for any graph F_n

$$|\mathcal{D}(F_n;\delta n)| \leq n! \cdot \binom{\frac{n^2}{2}}{\delta n} < n^{(\delta+1)n} 3^{\delta n}, \tag{20}$$

since we can permute the n vertices and change at most δn entries in the adjacency matrix $\mathbf{A}_F$ and we use the bound that $\binom{N}{k} \leq \frac{N^k}{k!} \leq (\frac{N}{k})^k 3^k$. Let $\mathcal{B}$ denote the set of graphs having exactly $\frac{cn}{2}$ edges and the size of $\mathcal{B}$ is

$$|\mathcal{B}| = \binom{\frac{n^2}{2}}{\frac{cn}{2}} \geq (\frac{n^2}{cn})^{cn/2} = (\frac{n}{c})^{cn/2}.$$

We can now bound the probability that a random graph $G_n \sim \mathcal{G}(n, c/n)$ belongs to set $\mathcal{D}(F_n;\delta n)$ for any given graph F_n is

$$\mathbb{P}[G_n \in \mathcal{D}(F_n;\delta n)] \leq \frac{\mathbb{P}[G_n \in \mathcal{D}(F_n;\delta n)]}{\mathbb{P}[G_n \in \mathcal{B}]}$$

$$\leq \frac{|\mathcal{D}(F_n;\delta n)| \max\limits_{g\in\mathcal{D}(F_n;\delta n)} \mathbb{P}[G_n=g]}{|\mathcal{B}| \min\limits_{g\in\mathcal{B}} \mathbb{P}[G_n=g]}$$

$$\overset{(a)}{\leq} \frac{|\mathcal{D}(F_n;\delta n)|}{|\mathcal{B}|} = n^{(\delta+1-c/2)n} 3^{\delta n},$$

where inequality (a) is due to the fact that $\min_{g\in\mathcal{B}} \mathbb{P}[G_n=g] \geq \max_{g\in\mathcal{S}(F_n)} \mathbb{P}[G_n=g]$ (i.e., the mode of the binomial distribution). Hence, $\mathbb{P}[G_n \in \mathcal{D}(F_n;\delta n)]$ decays to zero as $n\to\infty$, when $\delta < 0.5c-1$. □

Thus, for any given graph, a random graph does not have edit distance less than $(0.5c-1)n$ from it. It is thus reasonable to expect for any graph reconstruction algorithm to achieve an edit distance less than $(0.5c-1)n$, since otherwise, a random choice of the output graph could achieve the same edit distance. We now provide a lower bound on the fraction of the participating nodes such that no algorithm can reconstruct the original graph up to an edit distance less than $(0.5c-1)n$.

Theorem 4 (Lower Bound) *For $G_n \sim \mathcal{G}(n,c/n)$ and any set of participants V_n, for any graph estimator $\widehat{G}_n$ using (exact) shortest path distances between the participating node pairs, we have*

$$\mathbb{P}[\Delta(\widehat{G}_n, G_n; V) > \delta n] \to 1, \text{ when}$$
$$|V|^2 < Mn(0.5c-\delta-1)\frac{\log n}{\log\log n}, \tag{21}$$

for a small enough constant $M>0$ and any $\delta < (0.5c-1)$.

Thus, no algorithm can reconstruct G_n up to edit distance δn, for $\delta < 0.5c-1$, if the number of participating nodes is below a certain threshold. From Lemma 1, almost every random graph has an edit distance greater than $(0.5c-1)n$ from a given graph. Thus, when the number of participating nodes is below a certain threshold, accurate reconstruction by any algorithm is impossible.

Remarks:

(i) The lower bound does not require that the participating nodes are chosen uniformly and holds for any set of participating nodes of given cardinality.

(ii) The lower bound is analogous to a strong converse in information theory [61] since it says that the probability of edit distance being more a certain quantity goes to one (not just bounded away from zero).

(iii) The result is valid even for the scenario where second shortest path distances are used since the maximum second shortest path distance is also $O(\log n)$.

(iv) We have earlier shown that our algorithms RGD1 and RGD2 have good performance under a sub-linear number of participants. Closing the gaps in the exponents between lower bound and achievability is of interest.

Proof Ideas:

The proof is based on information-theoretic covering type argument, where cover the range of the estimator with random graphs of high likelihood. Using bounds on binomial distribution, we obtain the desired lower bound.

8.2 Non-Identifiability of General Topologies

Our proposed algorithms require the knowledge of shortest and second shortest path distances. Performance analysis reveals that the knowledge of second shortest path can greatly improve the

Figure 6: Example of two graphs with unit lengths where nodes a, b, u, v, w are participating. Even under all path length information between the participating nodes, the two graphs cannot be distinguished.

accuracy of topology discovery for random graphs. We now address the question if this can be accomplished in general.

To this end, we provide a counter-example in Fig.6, where a significant fraction of nodes are participating, and we are given distances along all the paths between the participants; yet, the topology cannot be correctly identified by any algorithm. This reveals a fundamental non-identifiability of general topologies using only a subset of participating nodes.

8.3 Relationship to Phylogenetic Trees

We note some key differences between the phylogenetic-tree model [49] and the additive delay model employed in this paper. In phylogenetic trees, sequences of extant species are available, and the unknown phylogenetic tree is to be inferred from these sequences. The phylogenetic-tree models the series of mutations occurring as the tree progresses and new species are formed. Efficient algorithms with low sample complexity have been proposed for phylogenetic-tree reconstruction, e.g., in [58, 62].

In the phylogenetic-tree model, the correlations along the phylogenetic tree decay exponentially with the number of hops. This implies that long-range correlations (between nodes which are far away) are "hard" to estimate, and require large number of samples (compared to the size of the tree) to find an accurate estimate. However, under the delay model, the delays are additive along the edges, and even long-range delays can be shown to be "easy" to estimate. Hence, the delay model does not require the more sophisticated techniques developed for phylogenetic-tree reconstruction (e.g., [62]), in order to achieve low sample complexity. However, the presence of cycles complicates the analysis for delay-based reconstruction of random graphs. Moreover, we developed algorithms when additional information is available in the form of second shortest-path distances. Such information cannot be obtained from phylogenetic data. We demonstrated that this additional information leads to drastic improvement in the accuracy of random-graph discovery.

9 Conclusion

In this paper, we considered discovery of sparse random graph topologies using a sub-linear number of uniformly selected participants. We proposed local quartet-based algorithms which exploit the locally tree-like property of sparse random graphs. We first showed that a sub-linear edit-distance guarantee can be obtained using end-to-end measurements along the shortest paths between a

sub-linear number of participants. We then considered the scenario where additionally, second shortest-path measurements are available, and showed that consistent topology recovery is feasible using only a sub-linear number of participants. Finally, we establish a lower bound on the edit distance achieved by any algorithm for a given number of participants. Our algorithms are simple to implement, computationally efficient and have low sample complexity.

There are many interesting directions to explore. Our algorithms require the knowledge of the bounds on the delay variances (i.e., edge lengths), and algorithms which remove these requirements can be explored. Our algorithms are applicable for other locally tree-like graphs as well, while the actual performance indeed depends on the model employed. Exploring how the reconstruction performance changes with the graph model is of interest. In many networks, such as peer-to-peer networks, there is a high churn rate and the nodes join and leave the networks, and it is of interest to extend our algorithms to such scenarios. Moreover, we have provided reconstruction guarantees in terms of edit distance with respect to the minimal representation, and plan to analyze reconstruction of other graph-theoretic measures such as the degree distribution, centrality measures, and so on. While we have assumed uniform sampling, other strategies (e.g., random walks) need to analyzed. We plan to implement the developed algorithms developed on real-world data.

Acknowledgements

The authors thank the anonymous reviewers for comments which significantly improved this paper. The first author is supported in part by the setup funds at UCI and the AFOSR Award FA9550-10-1-0310.

A Properties of Random Graphs

We first note the number of cycles in random graphs.

Lemma 2 (Cycles in Erdős-Rényi Random Graphs) *In $G_n \sim \mathcal{G}(n, \frac{c}{n})$, the expected number of cycles of lengths l is $O(c^l)$. Moreover, the number of two overlapping cycles of length l, denoted by H_l, satisfies*

$$\mathbb{E}[N_{H_l}] = O(n^{-1}c^{2l+1}). \tag{22}$$

Thus, there are a.a.s. no overlapping cycles of length less than $\frac{(1-\delta)\log n}{2\log c}$ for $\delta > 0$.

Proof: The proof is along the lines of [18, Cor. 4.9], but we specialize it for cycles. By counting argument, the expected number of cycles is given by

$$\mathbb{E}[N_{\mathcal{C}(l)}] = \binom{n}{l}\frac{l!}{2l}\left(\frac{c}{n}\right)^l = O(c^l).$$

Let number of vertices in H, be $|v(H_l)| = s$ with $l < s \leq 2l$. Note that the number of edges $|H_l| \geq s+1$ to be overlapping cycles. Hence,

$$\begin{aligned}\mathbb{E}[N_{H_l}] &\leq \binom{n}{s}(s!)\left(\frac{c}{n}\right)^{s+1} \\ &= O(n^{-1}c^{s+1}),\end{aligned}$$

and we obtain the desired result. □

Since we are dealing with the minimal representative $\widetilde{G}_n$ obtained by contracting nodes of degree < 3 in the original random graph $G_{n'}$, we need to derive its distribution. First note that $n = \Theta(n')$ a.a.s., where n' is the number of nodes in the original graph. On lines of [63, Lemma 4.4] and [63, Lemma 5.1], conditioned on n nodes in the minimal representative, the resulting graph is Erdős-Rényi, conditioned on the event that the minimum degree is at least three and denote this distribution as $\mathcal{G}'(n, \frac{c}{n})$.

We now obtain a lower bound on the size of the neighborhood in l hops in $\mathcal{G}'(n, c/n)$. Let $\Gamma_l(i)$ denote the set of nodes at graph distance l from node i in $\mathcal{G}'(n, \frac{c}{n})$. We have the following result.

Lemma 3 (Neighborhood in $\mathcal{G}'(n, \frac{c}{n})$) *For each node i in graph $\widetilde{G}_n \sim \mathcal{G}'(n, \frac{c}{n})$, with probability at least $1 - o(1/n)$,*

$$|\Gamma_l(i)| \geq \frac{1}{(\sqrt{c}-1)^2} c^{l-l_0} \log n, \tag{23}$$

for all $l_0 \leq l \leq \frac{3 \log n}{5 \log c}$, where

$$l_0 \leq \frac{\log \frac{9 \log n}{(\sqrt{c}-1)^2}}{\log 3}. \tag{24}$$

Proof: The proof is along the lines of [64, Lemma 6] but with modification to account for the minimum degree of three. Let l_0 denote the first time when

$$|\Gamma_{l_0}(i)| \geq \frac{9 \log n}{(\sqrt{c}-1)^2}. \tag{25}$$

Since the minimum degree is at least three, l_0 is given by (24). The rest of the proof proceeds along the lines of [64, Lemma 6]. □

We now provide bounds on the number cycles in $\mathcal{G}'(n, \frac{c}{n})$. Let

$$\xi(c) := 1 - e^{-c} - ce^{-c} - 0.5c^2 e^{-c}. \tag{26}$$

Lemma 4 (Cycles in $\mathcal{G}'(n, \frac{c}{n})$) *In $\widetilde{G}_n \sim \mathcal{G}'(n, \frac{c}{n})$, the expected number of cycles of lengths l is*

$$\mathbb{E}'[N_{C_l}] = O\left(\left(\frac{c}{\xi(c)}\right)^l\right). \tag{27}$$

Moreover, the number of two overlapping cycles of length l, denoted by H_l, satisfies

$$\mathbb{E}'[N_{H_l}] = O\left(\frac{n^{-1} c^{2l+1}}{\xi(c)^l}\right). \tag{28}$$

Proof: Let $\mathbb{E}'[N_{C_l}]$ denote the expected number of cycles of length l in random graph $\mathcal{G}'(n, c/n)$ and let $\mathbb{E}[N_{C_l}]$ denote the corresponding number in Erdős-Rényi random graph $\mathcal{G}(n, c/n)$. Let $\Lambda_n(l)$ denote the event that all given l nodes have degree at least three in $\mathcal{G}(n, c/n)$, and let $\Phi_n(l)$ denote the event that all given l nodes have degree at least three in $\mathcal{G}(n, c/n)$ and have edges only to nodes other than the given l nodes. Thus, we have that

$$\mathbb{E}'[N_{C_l}] = \mathbb{E}[N_{C_l}|\Lambda(l)] = \frac{\mathbb{E}[N_{C_l} 1_{\Lambda(l)}]}{\mathbb{P}[\Lambda(l)]} \leq \frac{\mathbb{E}[N_{C_l}]}{\mathbb{P}[\Lambda(l)]} = O\left(\left(\frac{c}{\xi(c)}\right)^l\right),$$

where 1 denotes indicator event and

$$\mathbb{P}[\Lambda_n(l)] \geq \mathbb{P}[\Phi_n(l)] = (\mathbb{P}[\Phi_n(1)])^l \overset{n\to\infty}{=} \xi(c)^l, \tag{29}$$

where the last result is from the fact that the asymptotic degree distribution of a node is the Poisson distribution. Similarly we have the other result on number of overlapping cycles. □

B Proof of Theorem 1

To prove the reconstruction guarantees for RGD1 algorithm, we first characterize "good" events which lead to accurate addition of edges in each step of RGD1 algorithm. We then bound the number of "bad" events which leads to an edit distance guarantee between the reconstructed graph $\widehat{G}$ by RGD1 algorithm (under exact distances) and the minimal representation of the original graph $\widetilde{G}$.

Recall in Section 6.1, we introduced the concept of bad quartets, where a middle hidden node is part of a cycle of length less than $2R'$ hops in the original graph G_n, where $R' = Rg/f$. Such quartets have wrong edge lengths or are not discovered. We weaken the criterion for bad quartets as those, where a middle hidden node is part of a (generalized) cycle of length less than $3R'$ hops. We note that this suffices to guarantee the presence of good witnesses which leads to accurate merging of the quartet under consideration. We prove this fact below.

Lemma 5 (Correctness of RGD1 for good quartets) *Given a minimal representation $\widetilde{G}$ and a set of observed nodes V, conditioned on the event that every edge in $\widetilde{G}$ is part of a short quartet (with edge lengths less than $Rg + \tau$), each short quartet is successfully and accurately merged by* RGD1 *when its middle hidden node is not part of a (generalized) cycle of length less than $3R'$ hops.*

Proof: The proof proceeds by induction on the steps of RGD1. Initially the graph is empty and since the quartet added is good, it is correct. At any step, assume that the graph $\widehat{G}$ is accurate (i.e., either the hidden nodes and paths are not yet added, or if there are added are correct). Let $Q(ab|uv)$ be the quartet to be merged with $\widehat{G}$ and let h_1 and h_2 be its two hidden nodes. If TreeMerge procedure is called by RGD1 algorithm in this step, it is accurate since it correctly adds the quartet $Q(ab|uv)$ to $\widehat{G}$. If CycleMerge procedure is called instead, the quartet $Q(ab|uv)$ is accurately merged if the join points between the existing paths in $\widehat{G}$ and the new paths to be created are correct. Note that the distance between hidden nodes h_1 and h_2 to be added and the join points to be inferred is at most R' hops. Since each of the join points is part of the short quartet, these short quartets are part of the witness set $\mathcal{W}$. If the witness quartets are not part of cycles of length less than $2R'$ hops, then they are guaranteed to be of the correct length and the join points for $Q(ab|uv)$ are correctly discovered. This implies that the middle nodes in $Q(ab|uv)$ are required to be not part of generalized short cycles of length less than $3R'$. Thus, the graph $\widehat{G}$ is accurate upon merging $Q(ab|uv)$. This implies the correctness of RGD1 at each step and thus, the above statement holds. □

Thus, the above result implies that the errors occur due to the following events: let $\mathcal{E}_1(e; \widetilde{G}_n, V_n)$ denote the event that the edge e is not a middle edge in any short quartet. Let $\mathcal{E}_2(v; \widetilde{G}_n, V_n)$ denote the event that the node v is the middle node of a bad short quartet, and let K_v denote the number of such bad short quartets (with participating nodes as end points and v as one of the middle

nodes). The edit distance satisfies

$$\Delta(\widehat{G}_n, \widetilde{G}_n; V_n) \leq \sum_{v \in \widetilde{G}_n} (\mathrm{Deg}(v; \widetilde{G}_n) + 6K_v)\, \mathbb{I}[\mathcal{E}_2(v)] + 2n^2 \sum_{e \in G_n} \mathbb{I}[\mathcal{E}_1(e)]. \tag{30}$$

This is because under event $\mathcal{E}_2(v)$, v is the middle node of a bad quartet, either it is not reconstructed, in which case, it contributes an edit distance of at most $\mathrm{Deg}(v)$, or the bad quartet is reconstructed with wrong edge lengths. In this case, it amounts to adding three wrong edges and not reconstructing the three correct edges. Thus, the edit distance is at most $6K_v$, where K_v is the number of bad quartets having v as a middle node under this event. For event $\mathcal{E}_1(e)$, where there is no short quartet containing e as a middle edge, we use the trivial bound on the edit distance as $2n^2$.

For the event $\mathcal{E}_1(e)$, we have

$$\mathbb{P}[\mathcal{E}_1(e; G_n, V_n)] \leq 2 \max_{v \in V} \mathbb{P}[|V_n \cap B_{R/2}(v; \widetilde{G}_n)| < 2],$$

since $\mathcal{E}_1^c(e; G_n, V_n)] = \{|V_n \cap B_{R/2}(v_1; \widetilde{G}_n) \geq 2\} \cap \{|V_n \cap B_{R/2}(v_2; \widetilde{G}_n) \geq 2\}$, where v_1 and v_2 are the endpoints of e. We now have

$$\mathbb{P}\left[|V_n \cap B_{R/2}(v; \widetilde{G}_n)| < 2 \middle| |B_{R/2}(v; \widetilde{G}_n)| \geq k\right] \leq (1-\rho)^k \left(1 + \frac{\rho}{1-\rho}\right).$$

We have a lower bound on $|B_{R/2}(v; \widetilde{G}_n)|$ from Lemma 3. Hence, for

$$R_{\min} \leq R \leq R_{\max}, \tag{31}$$

where $R_{\min}$ and $R_{\max}$ are given by (10), with probability $1 - o(n^{-1})$, we have

$$|B_{R/2}(v)| \geq (1-\delta) c^{(R - R_{\min})/2},$$

for some constant $\delta > 0$. Thus,

$$\mathbb{P}[|V \cap B_{R/2}(v)| < 2] \leq (1-\rho)^{(1-\delta)c^{(R-R_{\min})/2}} (1 + \frac{\rho}{1-\rho}). \tag{32}$$

For the second event $\mathcal{E}_2$, that the edge v is a part of a bad quartet, this occurs when it is part of a (generalized) cycle of length less than $3R'$,

$$\mathbb{P}[\mathcal{E}_2(v; \widetilde{G}_n, V_n)] = \mathbb{P}[v \in \mathcal{C}(3R'; \widetilde{G}_n)],$$

where $R' = Rg/f = \frac{\gamma g \log n}{f \log c}$. We have from Lemma 4,

$$\mathbb{P}[e \in \mathcal{C}(3R'; \widetilde{G}_n)] = O\left(\left(\frac{c}{\xi(c)}\right)^{3R'}\right).$$

The number of bad short quartets K_v satisfies

$$K_v = \tilde{O}(\rho^4 c^{2R'}),$$

since $K_v \leq |V_n \cap B_{R'/2}(v; \widetilde{G}_n)|^4$ and

$$\mathbb{E}[|V_n \cap B_{R'/2}(v; \widetilde{G}_n)|] \leq \rho \left(\frac{c}{\xi(c)}\right)^{R'/2},$$

and using Chernoff bounds, we have $|V_n \cap B_{R'/2}(v; G_n)| = \tilde{O}(\rho(c/\xi(c))^{R'/2})$ with probability $1 - o(n^{-1})$.

Thus, the expected edit distance is

$$\begin{aligned}\mathbb{E}[\Delta(\widehat{G}_n, \widetilde{G}_n; V_n)] &= O(n^4(1-\rho)^{(1-\delta)c^{(R-R_{\min})/2}}) \\ &\quad + \tilde{O}\left(\left(\frac{c}{\xi(c)}\right)^{5R'} \rho^4\right). \end{aligned} \tag{33}$$

Let $\rho_n = n^{-\beta}$. We have

$$(1-\rho)^{\Theta(c^{(R-R_{\min})/2})} = O(\exp[-n^{\gamma/2-\beta}]),$$

when $\gamma > 2\beta$ and $\gamma := \frac{\log c}{\log n}(R - R_{\min})$. When $(c/\xi(c))^{R'} = n^{\mu g/f}$, the second term in (33) is $\tilde{O}(n^{5\mu g/f - 4\beta})$, and is the dominant error event. Thus, the expected edit distance is

$$\mathbb{E}[\Delta(\widehat{G}_n, \widetilde{G}_n; V_n)] = \tilde{O}(n^{5\mu g/f - 4\beta}).$$

By Markov inequality, we have the result. □

C Proof of Theorem 2

The proof follows the lines of proof of Theorem 1. It is easy to note that each step of RGD2 succeeds and accurately merges a candidate quartet $Q(ab|uv)$ when the quartet has at most one (generalized) cycle of length less than $3R'$. This is because in this case, the join points of the quartet $Q(ab|uv)$ in $\widehat{G}$ can be inferred using shortest and second shortest paths. As in Theorem 1, we require that all edges be part of short quartets. Thus, We again have error events $\mathcal{E}_1$ and $\mathcal{E}_2$ which lead to a bound on edit distance (30). As before, let $\mathcal{E}_1(v; \widetilde{G}_n, V_n)$ denote the event that the node v is not a middle node in any short quartet and $\mathcal{E}_2(v)$ is the event that the node v is the middle node of a bad short quartet. However, now, the definition of a bad quartet is different: it occurs only when node v part of at least two overlapping generalized cycles, both of length less than $3R'$, and denote such structures as $H_{3R'}$.

The analysis for $\mathcal{E}_1$ is same as in proof of Theorem 1. For $\mathcal{E}_2$, from Lemma 4, we have,

$$\mathbb{P}[v \in H_{3R'}] = O(n^{-2}(\frac{c}{\xi(c)})^{6R'}).$$

Thus, the expected edit distance is

$$\begin{aligned}\mathbb{E}[\Delta(\widehat{G}_n, \widetilde{G}_n; V_n)] &= O(n(1-\rho)^{c^{(R-R_{\min})/2}}) \\ &\quad + O(n^{-1}(\frac{c}{\xi(c)})^{8R'}\rho^4).\end{aligned}$$

Thus, we have the desired result. □

D Proof of Theorem 3

The proof follows the lines of sample complexity results in [21]. From [21, Proposition 1], we have concentration bounds for delays (distances) under m samples as

$$\mathbb{P}[|\widehat{l}^m(i,j) - l(i,j)| > \epsilon] \leq 2\exp[-\frac{m\epsilon^2}{MR^3}],$$

for any $\epsilon > 0$, some constant $M > 0$, and for all $i, j \in V_n$. Taking union bound over all node pairs, we see that when $m = \Omega(\text{poly}(\log n))$, we have concentration of all the distances and we have the desired result. □

E Proof of Theorem 4

We use a covering argument for obtaining the lower bound, inspired by [65, Thm. 1]. For reconstructed graph $\widehat{G}$ using shortest path distances between $O(|V(G)|^2/2)$ node pairs, the range $\mathcal{R}(\widehat{G})$ of the estimator is bounded by

$$|\mathcal{R}(\widehat{G})| \leq (\text{Diam}(G))^{|V(G)|^2/2},$$

since the delay variances on the edges are assumed to be known exactly, and the shortest path can range from 1 to Diam(G). For $G \sim \mathcal{G}(n, c/n)$, the diameter[8] is $O(\log n)$ w.h.p. Let $\mathcal{S}(\widehat{G}; \delta n)$ denote all the graphs which are within edit distance of δn of the graphs in range $\mathcal{R}(\widehat{G})$

$$\mathcal{S}(\widehat{G}; \delta n) := \{F : \Delta(F, G') \leq \delta n, \text{ for some } G' \in \mathcal{R}(\widehat{G})\}.$$

Thus, using (19) and (20),

$$|\mathcal{S}(\widehat{G}; \delta n)| \leq \bigcup_{G' \in \mathcal{R}(\widehat{G})} |\mathcal{D}(G'; \delta n)| \leq |\mathcal{R}(\widehat{G})| n^{(\delta+1)n} 3^{\delta n}.$$

For the original graph $G \sim \mathcal{G}(n, c/n)$, we have the required probability

$$\begin{aligned}
\mathbb{P}[\Delta(\widehat{G}, G; V) > \delta n] &= \sum_{g \in \mathcal{S}^c} \mathbb{P}[\Delta(\widehat{G}, g; V)] > \delta n] \mathbb{P}(G = g) \\
&\quad + \sum_{g \in \mathcal{S}} \mathbb{P}[\Delta(\widehat{G}, g; V)] > \delta n] \mathbb{P}(G_n = g) \\
&\geq \sum_{g \in \mathcal{S}^c} \mathbb{P}[\Delta(\widehat{G}, g; V) > \delta n] \mathbb{P}(G_n = g) \\
&\overset{(a)}{=} \sum_{g \in \mathcal{S}^c} \mathbb{P}(G_n = g) \\
&\overset{(b)}{=} 1 - \sum_{g \in \mathcal{S}} \mathbb{P}(G_n = g), \qquad (34)
\end{aligned}$$

[8] The diameter of $G(n, \frac{c}{n})$ is $C(c) \log n$ [66], where $C(c) = \frac{1}{\log c} + \frac{2}{c} + O(\frac{\log c}{c^2})$ as $c \to \infty$.

where equality (a) is due to the fact that $\mathbb{P}[\Delta(\widehat{G}, g; V) > \delta n] = 1$ for all $g \in \mathcal{S}^c$ and (b) is due to $\sum_{g\in\mathcal{S}} \mathbb{P}(G = g) + \sum_{g\in\mathcal{S}^c} \mathbb{P}(G = g) = 1$. From (34), it suffices to provide an asymptotic upper bound for the term $\Upsilon := \sum_{g\in\mathcal{S}} \mathbb{P}(G = g)$. Furthermore, let $e_g \in \{1, \ldots, \binom{n}{2}\}$ denote the number of edges in the graph $g \in \mathcal{G}_n$. Then,

$$\mathbb{P}(G = g) = \left(\frac{c}{n}\right)^{e_g} \left(1 - \frac{c}{n}\right)^{\binom{n}{2}-e_g}. \tag{35}$$

We have the general result that for graphs $g_1, g_2 \in \mathcal{G}_n$

$$e_{g_1} \leq e_{g_2} \quad \Rightarrow \quad \mathbb{P}(G = g_1) \geq \mathbb{P}(G = g_2). \tag{36}$$

Define

$$z := \min\left\{ l \in \mathbb{N} : \sum_{k=1}^{l} \binom{\binom{n}{2}}{k} \geq |\mathcal{S}| \right\} \tag{37}$$

We obtain

$$z \leq O(\frac{|V|^2 \log\log n}{\log n}) + 0.5n(\delta + 1) + o(1).$$

Thus,

$$\begin{aligned}
\Upsilon &:= \sum_{g\in\mathcal{S}} \mathbb{P}(G = g) \\
&\leq \sum_{k=0}^{z} \binom{\binom{n}{2}}{k} \left(\frac{c}{n}\right)^{k} \left(1 - \frac{c}{n}\right)^{\binom{n}{2}-k} \\
&\overset{(a)}{\leq} \exp\left[-\frac{4}{nc} \Big(0.5n(0.5c - \delta - 1) - O\big(\frac{|V|^2 \log\log n}{\log n}\big) - o(1) \Big)^2 \right]
\end{aligned}$$

where inequality (a) follows from the fact that $\Pr(\mathrm{Bin}(N, q) \leq k) \leq \exp(-\frac{2}{Nq}(Nq - k)^2)$ for $k \leq Nq$ with the identifications $N = \binom{n}{2}$, $q = c/n$ and $k = z$, and that $\binom{n}{2} \geq (n/2)^2$. Finally, we observe from (a) that if $|V|^2 < Mn(0.5c - \delta - 1)\frac{\log n}{\log\log n}$ for small enough $M > 0$, then $\Upsilon \to 0$ as $n \to \infty$ and we obtain the required result. □

References

[1] A. Anandkumar, A. Hassidim, and J. Kelner, "Topology Discovery of Sparse Random Graphs With Few Participants," in *Proc. of ACM SIGMETRICS*, June 2011.

[2] S. Kandula, D. Katabi, and J. Vasseur, "Shrink: A Tool for Failure Diagnosis in IP Networks," in *Proc. of ACM SIGCOMM Workshop on Mining network data*, Philadelphia, PA, Aug. 2005.

[3] A. Motter and Y. Lai, "Cascade-based Attacks on Complex Networks," *APS Physical Review E*, vol. 66, no. 6, pp. 65–102, 2002.

[4] B. Eriksson, P. Barford, R. Nowak, and M. Crovella, "Learning Network Structure from Passive Measurements," in *Proc. of the ACM SIGCOMM conference on Internet measurement*, Kyoto, Japan, Aug. 2007.

[5] Y. Vardi, "Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data." *J. of the American Statistical Association*, vol. 91, no. 433, 1996.

[6] A. Anandkumar, C. Bisdikian, and D. Agrawal, "Tracking in a Spaghetti Bowl: Monitoring Transactions Using Footprints," in *Proc. of ACM SIGMETRICS*, Annapolis, Maryland, USA, June 2008.

[7] D. Alderson, H. Chang, M. Roughan, S. Uhlig, and W. Willinger, "The many facets of internet topology and traffic," *AIMS J. on Networks and Heterogeneous Media*, vol. 1, no. 4, p. 569, 2006.

[8] D. Shah and T. Zaman, "Detecting Sources of Computer Viruses in Networks: Theory and Experiment," in *Proc. of ACM Sigmetrics*, New York, NY, June 2010.

[9] S. Fortunato, "Community Detection in Graphs," *Physics Reports*, 2009.

[10] F. Wu, B. Huberman, L. Adamic, and J. Tyler, "Information Flow in Social Groups," *Physica A: Statistical and Theoretical Physics*, vol. 337, no. 1-2, pp. 327–335, 2004.

[11] D. Acemoglu, A. Ozdaglar, and A. ParandehGheibi, "Spread of (mis) information in social networks," *Games and Economic Behavior*, 2010.

[12] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore Art Thou r3579x?: Anonymized Social Networks, Hidden Patterns, and Structural Steganography," in *Proc. of ACM Intl. Conf. on World Wide Web*, Banff, Canada, May 2007.

[13] "mtrace– Print multicast path." ftp://ftp.parc.xerox.com/pub/net-research/ipmulti.

[14] M. Gunes and K. Sarac, "Resolving anonymous routers in Internet topology measurement studies," in *Proc. of IEEE INFOCOM*, 2008, pp. 1076–1084.

[15] B. Yao, R. Viswanathan, F. Chang, and D. Waddington, "Topology inference in the presence of anonymous routers," in *Proc. of IEEE INFOCOM*, 2003.

[16] J. Ni, H. Xie, S. Tatikonda, and Y. Yang, "Efficient and dynamic routing topology inference from end-to-end measurements," *Networking, IEEE/ACM Transactions on*, vol. 18, no. 1, pp. 123–135, 2010.

[17] M. Wainwright and M. Jordan, "Graphical Models, Exponential Families, and Variational Inference," *Foundations and Trends in Machine Learning*, vol. 1, no. 1-2, pp. 1–305, 2008.

[18] B. Bollobás, *Random Graphs*. Academic Press, 1985.

[19] M. Jovanović, F. Annexstein, and K. Berman, "Modeling peer-to-peer network topologies through small-world models and power laws," in *TELFOR*, 2001.

[20] M. Newman, D. Watts, and S. Strogatz, "Random graph models of social networks," *Proc. of the National Academy of Sciences of the United States of America*, vol. 99, no. Suppl 1, 2002.

[21] S. Bhamidi, R. Rajagopal, and S. Roch, "Network Delay Inference from Additive Metrics," *To appear in Random Structures and Algorithms, on Arxiv*, 2010.

[22] A. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[23] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *J. of Machine Learning Research*, vol. 11, pp. 985–1042, 2010.

[24] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney, "Statistical properties of community structure in large social and information networks," in *Proc. of WWW*, 2008, pp. 695–704.

[25] A. Clauset, C. Moore, and M. Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, vol. 453, no. 7191, pp. 98–101, 2008.

[26] "Internet Mapping Project," http://www.cheswick.com/ches/map/.

[27] "The Skitter Project," http://www.caida.org/tools/measurement/skitter/.

[28] "Cooperative Analysis for Internet Data Analysis, (CAIDA)," http://www.caida.org/tools/.

[29] R. Govindan and H. Tangmunarunkit, "Heuristics for Internet Map Discovery," in *IEEE INFOCOM*, Tel-Aviv, Israel, June 2000.

[30] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP Topologies with Rocketfuel," *IEEE/ACM Tran. on networking*, vol. 12, no. 1, pp. 2–16, 2004.

[31] Y. Shavitt and E. Shir, "DIMES: Let the internet measure itself," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 5, p. 74, 2005.

[32] Y. He, G. Siganos, and M. Faloutsos, "Internet Topology," in *Encyclopedia of Complexity and Systems Science*, R. Meyers, Ed. Springer, 2009, pp. 4930–4947.

[33] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting Positive and Negative Links in Online Social Networks," in *ACM WWW Intl. Conf. on World Wide Web*, 2010.

[34] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, "Inferring Networks of Diffusion and Influence," in *Proc. of the ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 2010.

[35] S. Myers and J. Leskovec, "On the Convexity of Latent Social Network Inference," in *Proc. of NIPS*, 2010.

[36] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network Tomography: Recent Developments," *Stat. Sc.*, vol. 19, pp. 499–517, 2004.

[37] F. Chung, M. Garrett, R. Graham, and D. Shallcross, "Distance realization problems with applications to Internet tomography," *J. of Comp. and Sys. Sc.*, vol. 63, no. 3, pp. 432–448, 2001.

[38] Z. Beerliova, F. Eberhard, T. Erlebach, A. Hall, M. Hoffmann, M. Mihal ak, and L. Ram, "Network Discovery and Verification," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, p. 2168, 2006.

[39] T. Erlebach, A. Hall, and M. Mihal'ak, "Approximate Discovery of Random Graphs," *Lecture Notes in Computer Science*, vol. 4665, p. 82, 2007.

[40] D. Achlioptas, A. Clauset, D. Kempe, and C. Moore, "On the bias of traceroute sampling: Or, power-law degree distributions in regular graphs," *J. ACM*, vol. 56, no. 4, 2009.

[41] M. Kurant, M. Gjoka, C. T. Butts, and A. Markopoulou, "Walking on a Graph with a Magnifying Glass," in *Proceedings of ACM SIGMETRICS '11*, San Jose, CA, June 2011.

[42] S. Khuller, B. Raghavachari, and A. Rosenfeld, "Landmarks in graphs," *Discrete Appl. Math.*, vol. 70, no. 3, pp. 217–229, 1996.

[43] L. Reyzin and N. Srivastava, "On the longest path algorithm for reconstructing trees from distance matrices," *Information Processing Letters*, vol. 101, no. 3, pp. 98–100, 2007.

[44] ——, "Learning and verifying graphs using queries with a focus on edge counting," *Lecture Notes in Computer Science*, vol. 4754, p. 285, 2007.

[45] H. Mazzawi, "Optimally Reconstructing Weighted Graphs Using Queries," in *Symposium on Discrete Algorithms*, 2010, pp. 608–615.

[46] S. Choi and J. Kim, "Optimal query complexity bounds for finding graphs," in *Proc. of annual ACM symposium on Theory of computing*, 2008, pp. 749–758.

[47] M. Shih and A. Hero, "Unicast inference of network link delay distributions from edge measurements," in *Proc. of IEEE ICASSP*, vol. 6, 2002, pp. 3421–3424.

[48] N. Duffield, J. Horowitz, F. Presti, and D. Towsley, "Multicast topology inference from end-to-end measurements," *Advances in Performance Analysis*, vol. 3, pp. 207–226, 2000.

[49] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge Univ. Press, 1999.

[50] M. Gomez-Rodriguez, E. Balduzzi, M. DE, and B. Schölkopf, "Uncovering the temporal dynamics of diffusion networks," in *Intl. Conf. on Machine Learning*, Bellevue, WA, 2011.

[51] T. Lappas, E. Terzi, D. Gunopulos, and H. Mannila, "Finding effectors in social networks," in *Proc. of ACM SIGKDD*, 2010, pp. 1059–1068.

[52] T. Snowsill, N. Fyson, T. De Bie, and N. Cristianini, "Refining causality: who copied from whom?" in *Proc. of ACM SIGKDD*, 2011, pp. 466–474.

[53] N. Alon and J. Spencer, *The probabilistic method*. Wiley-Interscience, 2000.

[54] J. Pearl, *Probabilistic Reasoning in Intelligent Systems—Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[55] G. Bunke *et al.*, "Inexact graph matching for structural pattern recognition," *Pattern Recognition Letters*, vol. 1, no. 4, pp. 245–253, 1983.

[56] E. Lehmann, *Theory of Point Estimation*. New York, NY: Chapman & Hall, 1991.

[57] H.-J. Bandelth and A. Dress, "Reconstructing the shape of a tree from observed dissimilarity data," *Adv. Appl. Math*, vol. 7, pp. 309–43, 1986.

[58] P. L. Erdős, L. A. Székely, M. A. Steel, and T. J. Warnow, "A few logs suffice to build (almost) all trees: Part ii," *Theoretical Computer Science*, vol. 221, pp. 153–184, 1999.

[59] T. Jiang, P. E. Kearney, and M. Li, "A polynomial-time approximation scheme for inferring evolutionary trees from quartet topologies and its application," *SIAM J. Comput.*, vol. 30, no. 6, pp. 1942–1961, 2001.

[60] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge Univ. Press, 1999.

[61] T. Cover and J. Thomas, *Elements of Information Theory*. John Wiley & Sons, Inc., 2006.

[62] C. Daskalakis, E. Mossel, and S. Roch, "Optimal phylogenetic reconstruction," in *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, 2006, pp. 159–168.

[63] I. Benjamini, G. Kozma, and N. Wormald, "The mixing time of the giant component of a random graph," *Arxiv preprint*, 2006.

[64] F. Chung and L. Lu, "The diameter of sparse random graphs," *Advances in Applied Mathematics*, vol. 26, no. 4, pp. 257–279, 2001.

[65] G. Bresler, E. Mossel, and A. Sly, "Reconstruction of Markov Random Fields from Samples: Some Observations and Algorithms," in *Intl. workshop APPROX Approximation, Randomization and Combinatorial Optimization*. Springer, 2008, pp. 343–356.

[66] D. Fernholz and V. Ramachandran, "The diameter of sparse random graphs," *Random Structures and Algorithms*, vol. 31, no. 4, pp. 482–516, 2007.