



University of Pennsylvania  
ScholarlyCommons

---

Statistics Papers

Wharton Faculty Research


---

2007

# Distorted Metrics on Trees and Phylogenetic Forests

Elchanan Mossel  
*University of Pennsylvania*

Follow this and additional works at: [http://repository.upenn.edu/statistics\\_papers](http://repository.upenn.edu/statistics_papers)

 Part of the [Bioinformatics Commons](#), [Computational Biology Commons](#), and the [Statistics and Probability Commons](#)

---

## Recommended Citation

Mossel, E. (2007). Distorted Metrics on Trees and Phylogenetic Forests. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4 (1), 108-116. <http://dx.doi.org/10.1109/TCBB.2007.1010>

This paper is posted at ScholarlyCommons. [http://repository.upenn.edu/statistics\\_papers/396](http://repository.upenn.edu/statistics_papers/396)  
For more information, please contact [repository@pobox.upenn.edu](mailto:repository@pobox.upenn.edu).

---

# Distorted Metrics on Trees and Phylogenetic Forests

## Abstract

We study distorted metrics on binary trees in the context of phylogenetic reconstruction. Given a binary tree  $T$  on  $n$  leaves with a path metric  $d$ , consider the pairwise distances  $\{d(u, v)\}$  between leaves. It is well known that these determine the tree and the  $d$  length of all edges. Here we consider distortions  $\hat{d}$  of  $d$  such that for all leaves  $u$  and  $v$  it holds that  $|d(u, v) - \hat{d}(u, v)| < f/2$  if either  $d(u, v) < M$  or  $\hat{d}(u, v) < M$ , where  $d$  satisfies  $f \leq d(e) \leq g$  for all edges  $e$ . Given such distortions we show how to reconstruct in polynomial time a forest  $T_1, \dots, T_\alpha$  such that the true tree  $T$  may be obtained from that forest by adding  $\alpha - 1$  edges and  $\alpha - 1 \leq 2^{-\Omega(M/g)n}$ .

Metric distortions arise naturally in phylogeny, where  $d(u, v)$  is defined by the log-det of a covariance matrix associated with  $u$  and  $v$ . When  $u$  and  $v$  are “far”, the entries of the covariance matrix are small and therefore  $\hat{d}(u, v)$ , which is defined by log-det of an associated empirical correlation matrix may be a bad estimate of  $d(u, v)$  even if the correlation matrix is “close” to the covariance matrix.

Our metric results are used in order to show how to reconstruct phylogenetic forests with small number of trees from sequences of length logarithmic in the size of the tree. Our method also yields an independent proof that phylogenetic trees can be reconstructed in polynomial time from sequences of polynomial length under the standard assumptions in phylogeny. Both the metric result and its applications to phylogeny are almost tight.

## Keywords

Markov processes, biology computing, evolution (biological), polynomials, trees (mathematics), distorted metrics, general Markov model, pairwise distances, phylogenetic forests, phylogenetic reconstruction, phylogeny, polynomial time, trees, binary trees, biological system modeling, character generation, computational complexity, helium, phylogeny, polynomials, reconstruction algorithms, sequences, upper bound, CFN, jukes-cantor, phylogenetics, distortion, forest, metric, tree, algorithms, animals, computational biology, genome, humans, models, genetic, mutation, phylogeny

## Disciplines

Bioinformatics | Computational Biology | Statistics and Probability

# Distorted metrics on trees and phylogenetic forests

ELCHANAN MOSSEL \*

## Abstract

We study distorted metrics on binary trees in the context of phylogenetic reconstruction. Given a binary tree  $T$  on  $n$  leaves with a path metric  $d$ , consider the pairwise distances  $\{d(u, v)\}$  between leaves. It is well known that these determine the tree and the  $d$  length of all edges. Here we consider distortions  $\hat{d}$  of  $d$  such that for all leaves  $u$  and  $v$  it holds that  $|d(u, v) - \hat{d}(u, v)| < f/2$  if either  $d(u, v) < M$  or  $\hat{d}(u, v) < M$ , where  $d$  satisfies  $f \leq d(e) \leq g$  for all edges  $e$ . Given such distortions we show how to reconstruct in polynomial time a forest  $T_1, \dots, T_\alpha$  such that the true tree  $T$  may be obtained from that forest by adding  $\alpha - 1$  edges and  $\alpha - 1 \leq 2^{-\Omega(M/g)} n$ .

Metric distortions arise naturally in phylogeny, where  $d(u, v)$  is defined by the log-det of a covariance matrix associated with  $u$  and  $v$ . When  $u$  and  $v$  are “far”, the entries of the covariance matrix are small and therefore  $\hat{d}(u, v)$ , which is defined by log-det of an associated empirical-correlation matrix may be a bad estimate of  $d(u, v)$  even if the correlation matrix is “close” to the covariance matrix.

Our metric results are used in order to show how to reconstruct phylogenetic forests with small number of trees from sequences of length logarithmic in the size of the tree. Our method also yields an independent proof that phylogenetic trees can be reconstructed in polynomial time from sequences of polynomial length under the standard assumptions in phylogeny. Both the metric result and its applications to phylogeny are almost tight.

## 1 Introduction

Reconstructing phylogenies have been a scientific challenge for the last 50 years. We refer the reader to [9] or [16] for general and mathematical background. The standard setting in phylogeny is of trees where the leaves are labeled by *taxa* or *species*. Given aligned sequences at the leaves, we define character  $i$  to be the collection of letters at position  $i$  for all the species. Under the i.i.d. assumption, the characters are independent samples from the evolutionary process on the tree.

The theoretical foundations of most methods used in phylogeny are unsatisfactory. Under the standard i.i.d. model, *Parsimony* is not consistent [2, 8] and is NP hard to compute [4, 10, 11]. The computational complexity of finding the *Maximum likelihood* tree is not known and the best bounds on the the amount of data needed are *exponential* in the number of taxa [19].

Computational complexity and information theory considerations have not played an important role in phylogeny in the past as biologists were mostly interested in reconstructing trees on a small number (typically at most a few hundred) species. However, one of the major goals of systematic biology in the coming decade is to reconstruct phylogenies on millions of species. It is clear that

---

\*mossel@stat.berkeley.edu Miller fellow, CS and Statistics, U.C. Berkeley

for such numbers, it is crucial to apply algorithms with low computational complexity. Similarly, algorithms should use information efficiently.

In [5] the authors developed the first reconstruction algorithm satisfying two important properties:

- Given number of characters that is polynomial in the number of taxa, the algorithm finds the true tree with high probability.
- The running time of the algorithm is polynomial in the number of taxa.

Variants of these method and generalizations from two states models to general models appeared in [6]. In [3] the authors discuss a closely related problem of learning a phylogenetic tree (in the PAC setting). They developed a PAC learning algorithm for the two state model. The problem of PAC-learning general-state model in polynomial time is still open. See also [7] for an earlier result on learning phylogenies.

The method developed in [5] is a *distance* method. Such methods were commonly used in phylogeny before, but [5] is the result where a distance method yields a provably good performance. Distance methods are based on defining a path metric on the tree based on the evolution model. Then the distance between leaves of the tree is approximated by some distance between the corresponding sequences at the leaves. In this sense all distance methods in phylogeny may be view as reconstruction methods from distorted metrics.

Given the existence of a polynomial time reconstruction algorithms, the next problem is optimizing the sampling complexity. The number of characters needed (= the length of sequences) is of great practical importance, as this number is bounded by the underlying biology. It is therefore desirable to minimize this number.

Since there are  $2^{\Theta(n \log n)}$  binary trees on  $n$  leaves, an easy counting argument yields that the number of characters needed is at least logarithmic in the number of taxa. Thus we are led to the following natural problem: is the length of the sequences needed logarithmic in  $n$  as or is it polynomial?

In [14] we showed that for a restricted family of models, it is possible to reconstruct phylogenies from a logarithmic number of characters, if the mutation rates are low (bounded above by some constant). We also showed that a polynomial number of characters is needed if the mutation rates are high (bounded below by some constant).

We later [13] (see also [17]) generalized the polynomial lower bound for high mutation rates to a large family of models. In [15] we analyze another model where logarithmic reconstruction is achievable for low mutation rates.

The phase transition discussed above is of crucial interest if we wish to reconstruct all the tree. However, in some cases, a more modest objective is posed: reconstruct a “large portion” of the tree. Practitioners (this was kindly noted to me by J. Felsenstein (2001, private communication) and J. Kim (2003, private communication)) have noticed that this problem seems to be much easier than the problem of reconstructing the complete tree.

In this paper we prove that this is indeed the case.

**Definition 1.1.** *We define the operation of edge adding to a forest as one of the following*

- *Add an edge  $(u, v)$  connecting two isolated leaves  $u$  and  $v$ .*

- Given an edge  $(u, v)$  of the forest and an isolated leaf  $w$ , replace the edge  $(u, v)$  by the edges  $(u, w')$ ,  $(w', v)$  and  $(w', w)$  where  $w'$  is a new vertex.
- Replace the two edges  $(u_1, v_1)$  and  $(u_2, v_2)$  of the forest by  $(u_1, w_1)$ ,  $(w_1, v_1)$ ,  $(u_2, w_2)$ ,  $(w_2, v_2)$  and  $(w_1, w_2)$  where  $w_1, w_2$  are new vertices.

See figure 1.

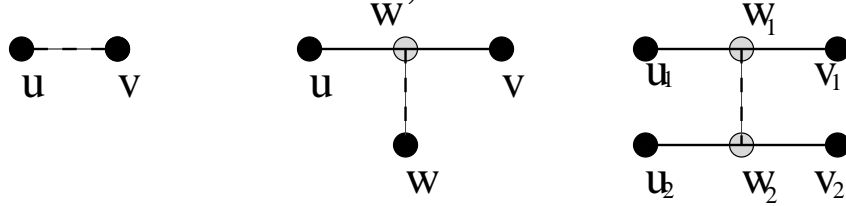


Figure 1: The “edge-adding” operation

We show that under the standard assumptions in phylogeny, for a tree  $T$  on  $n$  leaves and for all  $\delta > 0$  and  $\gamma > 0$  we can reconstruct from  $\gamma^{-O_\delta(1)} \log n$  characters a forest  $T_1, \dots, T_\alpha$  such that  $\alpha \leq 1 + \gamma n$  and that  $T$  may be obtained from the forest  $T_1, \dots, T_\alpha$  by adding at most  $\alpha - 1$  edges. The reconstruction is performed in a polynomial time and with error bounded by  $\delta$ .

Note that taking  $\gamma$  to be a small constant we obtain that “most” edges of the tree can be reconstructed from  $O(\log n)$  characters. Taking  $\gamma = 1/n$ , we obtain that the full tree may be reconstructed from a polynomial number of characters. Thus obtaining an independent proof of the results of [5, 6, 3].

Our results indicate what level of refinement is achievable in reconstructing a phylogenetic tree given a certain amount of data. We believe that our techniques may also play an important role in reconstructing the complete phylogenetic tree in low mutation rates from logarithmic number of characters as it allows a very clean divide and conquer approach (see [14]).

The main ingredient of the proof is a metric theorem that can be stated roughly as follows. Let  $T$  be a binary tree and  $d$  a path metric on  $T$  such that  $f \leq d(e) \leq g$  for all edges  $e$ . Let  $\mathcal{L}(T)$  be the set of leaves of  $T$  and  $\hat{d} : \mathcal{L}(T) \times \mathcal{L}(T) \rightarrow \mathbb{R}$  a distortion of  $d$  that satisfies  $|d(u, v) - \hat{d}(u, v)| < f/2$  if  $\hat{d}(u, v) < M$  or  $d(u, v) < M$  ( $\hat{d}$  typically does not correspond to a path metric on  $T$ ). We show that we can partition  $\mathcal{L}(T)$  into  $\alpha$  sets  $L_1, \dots, L_\alpha$ , where  $\alpha = n2^{-\Omega(M/g)}$ . For  $1 \leq \beta \leq \alpha$  we reconstruct a tree such that the leaves of  $T_\beta$  are  $L_\beta$  and the tree  $T$  may be obtained from the forest  $(T_\beta)_{\beta \leq \alpha}$  by adding at most  $\alpha - 1$  edges.

It is easy to see that the metric theorem is tight up to the constant in the  $\Omega$  by considering an  $r$ -level 3-regular tree, where the length of all edges is exactly  $g$  and  $\hat{d}(u, v) = \infty$  if  $d(u, v) > M$ . Similar tightness results hold for the number of trees in the forest in the phylogenetic reconstruction. The proof is more complicated and follows ideas from [13] on lower bounds on the sampling complexity of phylogenies. The proof is omitted in this extended abstract. Our metric result may be of independent interest to other problems where path metrics on trees are considered.

We now give a high level sketch of the different sections of the paper.

- The formal definition of the model and the statement of the main results are given in Section 2, where we also discuss how the metric result implies the result in phylogeny.

The reduction from the phylogenetic model to distorted metrics is given in Proposition 2.2 which is an easy reformulation of a large deviation result from [6]. This proposition gives  $M$  as a function of the number of samples. The distortion error is assumed to be at most  $\epsilon = f/2$ . It is known that Proposition 2.2 is essentially tight. In other words, distances larger than  $M$  are likely to be computed with error larger than  $f/2$ .

- Given the distortion, it is easy to construct for each leaf  $v$ , a tree  $T_v$  on the set of leaves in the  $(M - 7f)/6$  (metric)-neighborhood of  $v$ . This is done using standard techniques in Phylogeny.
- The collection of trees  $\{T_v\}$  is *not* the forest we are looking for. First, the trees in this forest are *not* disjoint. Second, there may be many trees in this collection (in fact as many as  $n$  different trees). The main task of the paper is to “glue” these trees to form *edge-disjoint* forest. Then we can bound the number of trees in the forest.
- The notion of edge-disjoint trees is studied in Section 3. The results of this section imply that if a forest of edge-disjoint trees is a refinement of the collection  $\{T_v\}$  above then the size of the forest is  $1 + n2^{-\Omega(M/g)}$ .
- The “glueing” algorithm is given in Section 4.
- The final metric result is stated in Section 5.

**Acknowledgments:** The idea that reconstructing a forest should be “easy” was conceived during a talk by Junhyong Kim at the kickoff meeting of the Cipres project. J. Kim said that it seems like most edges of phylogenies are easy to reconstruct. I thank him and Tandy Warnow for encouragement to work on this problem.

## 2 Definitions and main results

Let  $T$  be a tree. Write  $\mathcal{V}(T)$  for the nodes of  $T$ ,  $\mathcal{E}(T)$  for the edges of  $T$  and  $\mathcal{L}(T)$  for the leaves of  $T$ . If the tree is rooted, then we denote by  $\rho(T)$  the root of  $T$ . Unless stated otherwise, all trees are assumed to be *binary* (all internal degrees are 3) and it is further assumed that  $\mathcal{L}(T)$  is labeled.

Let  $T$  be a tree equipped with a path metric  $d : \mathcal{E}(T) \rightarrow \mathbb{R}_+$ .  $d$  will also denote the induced metric on  $\mathcal{V}(T)$ :

$$d(v, w) = \sum \{d(e) : e \in \text{path}(v, w)\}, \quad (1)$$

for all  $v, w \in \mathcal{V}(T)$ .

We will further assume below that the length of all edges is bounded between  $f$  and  $g$  for all  $e \in \mathcal{E}(T)$ ,

$$f \leq d(e) \leq g. \quad (2)$$

In applications to phylogeny we are typically given a distortion  $\hat{d} : \mathcal{L}(T) \times \mathcal{L}(T) \rightarrow \mathbb{R}_+$  of  $d$ . We define an  $(\epsilon, M)$  distortion as follows.

**Definition 2.1.** *Given a tree  $T$  equipped with a metric  $d$ , and two positive numbers  $0 < \epsilon < M$ , we say that  $\hat{d} : \mathcal{L}(T) \times \mathcal{L}(T) \rightarrow \mathbb{R}_+ \cup \{\infty\}$  is an  $(\epsilon, M)$  distortion of  $d$  if*

- $\hat{d}(u, v) = \hat{d}(v, u)$  for all  $u$  and  $v$  in  $\mathcal{L}(T)$ ; i.e.,  $\hat{d}$  is symmetric.
- If  $\hat{d}(u, v) = \infty$ , then  $d(u, v) > M$ .
- If  $\hat{d}(u, v) < \infty$ , then  $|\hat{d}(u, v) - d(u, v)| < \epsilon$ .

It is well known that  $d : \mathcal{L}(T) \times \mathcal{L}(T) \rightarrow \mathbb{R}_+$  determines the underlying tree  $T$  and the metric on the edges  $d : \mathcal{E}(T) \rightarrow \mathbb{R}_+$ . Moreover, there exists a polynomial time algorithm to reconstruct  $T$ . Similarly, we may recover  $T$  from any  $(\epsilon, \infty)$  distortion of  $d$  if  $\epsilon < f/2$ . Moreover, in this case, we may also recover a function  $\hat{d} : \mathcal{E}(T) \rightarrow \mathbb{R}$  satisfying  $|\hat{d}(e) - d(e)| < 2\epsilon$  ([1, 12], see e.g. [16, Chapter 7]).

In our main result we show that given an  $(\epsilon, M)$  distortion of  $d$ , we may recover many of the edges of  $T$  and a good approximation of the  $d$  length of those edges.

**Theorem 2.1.** *Let  $T = (V, E)$  be a binary tree equipped with a metric  $d$  satisfying (2). Let  $n = |\mathcal{L}(T)|$ , so that  $|\mathcal{V}(T)| = 2n - 2$ . Let  $\hat{d}$  be an  $(\epsilon, M)$  distortion of  $d$  and suppose that  $\epsilon < f/2$  and that  $M > 7\epsilon$ . Then  $\hat{d}$  determines a partition  $\mathcal{P}$  of  $\mathcal{L}(T)$  into sets  $L_1, \dots, L_\alpha$  and a forest  $T_1, \dots, T_\alpha$  such that  $L_\beta = \mathcal{V}(T_\beta) \cap \mathcal{L}(T)$  for all  $\beta$  and*

- The tree  $T$  may be obtained from the forest  $T_1, \dots, T_\alpha$  by adding at most  $\alpha - 1$  edges.
- The number of trees  $\alpha$  in the forest is at most  $\lfloor 1 + \frac{60n}{\sqrt{2}} 2^{-\frac{M-\epsilon}{2g}} \rfloor$ .

Moreover,

- the partition  $(L_\beta)_{\beta \leq \alpha}$ ,
- the trees  $(T_\beta)_{\beta \leq \alpha}$  and
- a function  $\hat{d} : \cup_{\beta \leq \alpha} \mathcal{E}(T_\beta) \rightarrow \mathbb{R}_+$  satisfying  $|\hat{d}(e) - d(e)| < 2\epsilon$ ,

can be all computed from  $\hat{d}$  in time polynomial on  $n$ .

**Mutation models and distances.** When reconstructing phylogenies, the data is given as sequences at the labeled leaves  $\mathcal{L}(T)$  and the tree  $T$  is unknown. Usually the mutation model is defined on a rooted tree while the goal is to reconstruct un-rooted trees (in many models there is no way to distinguish a root).

We let  $\mathcal{A}$  denote the alphabet in which information is encoded. For example,  $\mathcal{A} = \{A, C, G, T\}$  for DNA sequences,  $\mathcal{A} = \{20 \text{ amino acids}\}$  for proteins and  $\mathcal{A} = \{0, 1\}$  for purine-pyrimidine sequences. To define the mutation model we assume that all the edges are directed away from the root and for each edge  $e \in \mathcal{E}(T)$  let  $M(e)$  be the mutation matrix corresponding to the edge  $e$ .  $M(e)$  is an  $|\mathcal{A}| \times |\mathcal{A}|$  stochastic matrix. The  $(i, j)$ 'th entry of  $M(e)$  is the probability that state  $i$  will mutate to state  $j$  along edge  $e$ . It is assumed that each character evolves down the tree as a Markov-chain on the tree, where  $M(e)$  is the transition matrix for edge  $e$ . The root letter is chosen from some fixed distribution  $\pi$ . It is assumed that the characters evolve in an i.i.d. manner - they all come from the same distribution and each one is independent from all the others.

Two popular examples are the CFN model where  $M(e) = \begin{pmatrix} 1 - \theta(e) & \theta(e) \\ \theta(e) & 1 - \theta(e) \end{pmatrix}$  and the Jukes-Cantor models where

$$M(e) = \begin{pmatrix} 1 - 3\theta(e) & \theta(e) & \theta(e) & \theta(e) \\ \theta(e) & 1 - 3\theta(e) & \theta(e) & \theta(e) \\ \theta(e) & \theta(e) & 1 - 3\theta(e) & \theta(e) \\ \theta(e) & \theta(e) & \theta(e) & 1 - 3\theta(e) \end{pmatrix}.$$

It turns out that under mild assumptions on the matrices  $M(e)$  and the evolution model - the *log-det* distance defines a path metric on the tree [18]. We summarize the basic properties of this distance in the following proposition. The proof for the CFN and the Jukes-Cantor model have appeared independently several times. The general case follows from a large deviation estimate in [6] and is proven in the appendix.

**Proposition 2.2.** *Assume that the matrices  $M(e)$  satisfy that  $e^{-2g'} < \det(M(e)) < e^{-2f}$  for all  $e \in \mathcal{E}(T)$  and that for all nodes  $v \in \mathcal{V}(T)$  and all letters  $a \in \mathcal{A}$ , the probability that the letter at  $v$  is  $a$  is at least  $\pi_{\min} > 0$ . Let  $\epsilon > 0$ .*

*For every two vertices  $u, v \in \mathcal{L}(T)$  and  $a, b \in \mathcal{A}$ , let  $F_{a,b}$  be the probability that node  $u$  has letter  $a$  and node  $v$  has letter  $b$ . Let  $\hat{F}_{a,b}$  be the empirical distribution that node  $u$  has letter  $a$  and node  $v$  has letter  $b$ . Let  $d(u, v) = -\log \det(F_{i,j})$  and  $\hat{d}(u, v) = -\log \det(\hat{F}_{i,j})$  if  $-\log \det(\hat{F}_{i,j}) \leq M + \epsilon$ , and  $\hat{d}(u, v) = \infty$  if  $-\log \det(\hat{F}_{i,j}) > M + \epsilon$  or  $\det(\hat{F}_{i,j}) \leq 0$ . Then*

- $d(u, v)$  is a path metric on the tree satisfying  $g \geq d(e) \geq f$  for all edges  $e$  of the tree (where  $g$  depends on  $g'$  and  $\pi_{\min}$ ).
- There exists a constant  $c$  such that for all  $r > 2$  if the number of sample satisfies

$$k \geq \frac{cr}{(1 - e^{-2\epsilon})^2} e^{2M+2\epsilon} \log n, \quad (3)$$

*then with probability at least  $1 - n^{2-r}$  it holds that  $\hat{d}$  is an  $(\epsilon, M)$  distortion of  $d$ .*

The proposition may be used with different values of the parameters  $k$  and  $M$ . Fix  $\epsilon < f/2$ . Taking  $M$  to be the diameter of the tree, it gives that all empirical distances are within  $\epsilon$  of the true distances once  $k$  is exponential in  $M$ . Since  $M$  may be as large as  $\Omega(gn)$ , this gives sampling complexity  $k$  which is exponential in  $n$ . Taking  $M = 100g$ , say, would give an  $(\epsilon, 100g)$  distortion from  $k = O(e^{100g} \log n) = O(\log n)$  samples. In sequel we will use  $M$  ranging between  $M = O(g)$  for a constant  $c$  to  $k = O(g \log n)$  (in particular, typically we will only have a fraction of the distances within  $\epsilon$  of their true value).

Combining Proposition 2.2 and Theorem 2.1 we obtain

**Theorem 2.3.** *Consider a binary phylogenetic tree  $T$ , where the log-det distance associated with the mutation matrices  $M(e)$  satisfy the conditions of Proposition 2.2. Then given  $k$  satisfying (3), with  $M > 7\epsilon$ , and  $\epsilon < f/2$  we can with probability at least  $1 - n^{2-r}$  recover a partition  $\mathcal{P}$  of  $\mathcal{L}(T)$  into sets  $L_1, \dots, L_\alpha$  and a forest  $T_1, \dots, T_\alpha$  such that  $L_\beta = \mathcal{V}(T_\beta) \cap \mathcal{L}(T)$  for all  $\beta$  and*

- $T_1, \dots, T_\alpha$  is a forest that may be obtained from  $T$  by removing  $\alpha - 1$  edges.
- The number of trees  $\alpha$  in the forest is at most  $\lfloor 1 + \frac{60n}{\sqrt{2}} 2^{-\frac{M-\epsilon}{2g}} \rfloor$ .



Moreover, we can recover a function  $\hat{d} : \cup_{\beta \leq \alpha} \mathcal{E}(T_\beta) \rightarrow \mathbb{R}_+$  satisfying  $|\hat{d}(e) - d(e)| < 2\epsilon$  for all  $e$ .

Note that taking  $M = O(g)$  to be a large constant and  $\epsilon = f/2$  proves that most edges of the tree can be recovered from  $O(\log n)$  characters. Similarly, taking  $M = O(g \log n)$  and  $\epsilon = f/2$ , we see that we can recover the underlying tree from  $k = n^{O(g)}$  characters, thus obtaining an independent proof of the results of [5, 6, 3].

### 3 Edge disjoint trees

*Edge disjoint trees* and *edge sharing trees* will play a crucial role below. In this section we define these notions and discuss some of their basic properties.

We let  $T$  be a binary tree with vertices  $\mathcal{V}(T)$  and edges  $\mathcal{E}(T)$ . We let  $\mathcal{L}(T) \subset \mathcal{V}(T)$  be the set of leaves of  $T$  and  $n = |\mathcal{L}(T)|$  the size of this set. We write  $\text{path}_T(x, y)$  for the path (sequence of edges) connecting  $x$  to  $y$  in  $T$ . We will sometime omit the subscript  $T$  and write  $\text{path}(x, y)$ . We write  $\ell(x, y)$  or  $\ell_T(x, y)$  for the number of edges in the path connecting  $x$  and  $y$ . For two sets  $A, B \subset \mathbf{V}(T)$  we write  $\ell_T(A, B) = \min\{\ell(x, y) : x \in A, y \in B\}$ .

Removing an edge  $e$  from a tree  $T$  results in obtaining two trees  $T_1$  and  $T_2$ . The *split defined by*  $e$  is the partition  $\{\mathcal{L}(T) \cap \mathcal{V}(T_1), \mathcal{L}(T) \cap \mathcal{V}(T_2)\}$  of  $\mathcal{L}(T)$ . We denote by  $\Sigma(T)$  the collection of  $\mathcal{L}(T)$  splits defined by all edges of  $T$ . It is well know that  $\Sigma(T)$  determines  $T$  (see e.g. [16, Chapter 3]). We denote the split  $\{A, B\}$  by  $A|B$ .

**Definition 3.1.** *Let  $T$  be a binary tree and  $L \subset \mathcal{L}(T)$  a set of leaves. The restriction of  $T$  to  $L$  is defined as follows. This is the tree whose leave set is  $L$  and whose splits are defined by*

$$\Sigma(T|L) := \{A|A' : A = B \cap L, A' = B' \cap L \text{ and } B|B' \in \Sigma(T)\}.$$

The restriction of  $T$  to  $L$  is denoted by  $T|L$ .

Given two sets  $L_1, L_2 \subset \mathcal{L}(T)$ , we say that the trees  $T|L_1, T|L_2$  are *edge disjoint* if

$$\text{path}_T(u_1, v_1) \cap \text{path}_T(u_2, v_2) = \emptyset,$$

for all  $u_1, v_1 \in L_1$  and  $u_2, v_2 \in L_2$ . We say that  $T|L_1, T|L_2$  are *edge-sharing* if they are not edge disjoint.

The following easy lemma is useful as it shows that edge disjointness does not depend on the ambient tree.

**Lemma 3.1.** *Let  $L_1 \cup L_2 \subset L' \subset \mathcal{L}(T)$ . Then  $T|L_1$  and  $T|L_2$  are edge disjoint if and only if  $\text{path}_{T|L'}(u_1, v_1) \cap \text{path}_{T|L'}(u_2, v_2) = \emptyset$ , for all  $u_1, v_1 \in L_1$  and  $u_2, v_2 \in L_2$ .*

*In particular,  $L_1$  and  $L_2$  are edge disjoint if and only if  $\text{path}_{T|L_1 \cup L_2}(u_1, v_1) \cap \text{path}_{T|L_1 \cup L_2}(u_2, v_2) = \emptyset$ , for all  $u_1, v_1 \in L_1$  and  $u_2, v_2 \in L_2$ .*

*Proof.* The second statement follows immediately from the first one by letting  $L' = L_1 \cup L_2$ .

For the first statement, note that  $\text{path}_T(u_1, v_1)$  is obtained from  $\text{path}_{T|L'}(u_1, v_1)$  by replacing each edge  $(x, y) \in \text{path}_{T|L'}(u_1, v_1)$  by a sequence of edges  $(x = x_1, x_2), \dots, (x_j, x_{j+1} = y)$ . Moreover, the sequence  $(x = x_1, x_2), \dots, (x_j, x_{j+1} = y)$  depends on the edge  $(x, y)$  only and each edge  $(x_i, x_{i+1})$  of  $T$  appears in the sequence of at most one edge  $(x, y)$  of  $T|L'$ .

It now follows that  $\text{path}_{T|L'}(u_1, v_1) \cap \text{path}_{T|L'}(u_2, v_2) = \emptyset$ , for all  $u_1, v_1 \in L_1$  and  $u_2, v_2 \in L_2$  if and only if  $\text{path}_T(u_1, v_1) \cap \text{path}_T(u_2, v_2) = \emptyset$  for all  $u_1, v_1 \in L_1$  and  $u_2, v_2 \in L_2$ , as needed. ■

Next we state a useful closure property.

**Lemma 3.2.** *If  $T|L_1$  and  $T|L_2$  are edge sharing and  $L_1 \cup L_2 \subset \mathcal{L}(T)$ , then every edge of  $e$  of  $T|L_1 \cup L_2$  belongs to a path  $\text{path}_{T|L_1 \cup L_2}(u, v)$  where  $u, v \in L_1$  or  $u, v \in L_2$ .*

*Proof.* Suppose otherwise and let  $e = (w, w')$  be an edge of  $T|L_1 \cup L_2$  that does not belong to any such path. It follows that all the vertices of  $L_1$  are on one side of that edge and all the vertices of  $L_2$  on the other side.

Thus  $\text{path}_{T|L_1 \cup L_2}(u_1, v_1) \cap \text{path}_{T|L_1 \cup L_2}(u_2, v_2) = \emptyset$  for all  $u_1, v_1 \in L_1$  and  $u_2, v_2 \in L_2$ . This in turn implies by Lemma 3.1 that  $L_1$  and  $L_2$  are edge disjoint in contradiction to our assumption. ■

**Lemma 3.3.** *Suppose that  $(T|L_1, T|L_2)$  are edge sharing while  $(T|L_1, T|L_3)$  and  $(T|L_2, T|L_3)$  are edge disjoint, and let  $L = L_1 \cup L_2$ . Then  $(T|L, T|L_3)$  are edge disjoint.*

*Proof.* Suppose otherwise. Let  $u, u' \in L$  and  $v, v' \in L_3$  such that  $\text{path}_{T|L}(u, u') \cap \text{path}_{T|L}(v, v') \neq \emptyset$ . Let  $e$  be an edge that belongs to their intersection. By the previous lemma, it follows that there exists  $w, w' \in L_1$  or  $w, w' \in L_2$  such that  $e \in \text{path}_{T|L}(w, w')$ . Now  $\text{path}_{T|L}(w, w') \cap \text{path}_{T|L}(v, v')$  is not empty - in contradiction to the fact that  $(T|L_1, T|L_3)$  and  $(T|L_2, T|L_3)$  are edge disjoint. ■

We note that for binary trees, the notions of edge disjointness and vertex disjointness coincide. Let  $T$  be a tree and  $L_1, L_2 \subset \mathcal{L}(T)$ . We say that  $T|L_1$  and  $T|L_2$  are *vertex-disjoint* if  $\text{path}_T(u_1, v_1)$  and  $\text{path}_T(u_2, v_2)$  have no vertices in common for all  $u_1, v_1 \in L_1$  and  $u_2, v_2 \in L_2$ .

**Proposition 3.4.** *Let  $T$  be a tree and let  $L_1, L_2 \subset \mathcal{L}(T)$ . Then if  $T|L_1$  and  $T|L_2$  are vertex-disjoint, they are also edge-disjoint*

*Let  $T$  be a tree where all the internal nodes are of degree 2 or 3 and let  $L_1, L_2 \subset \mathcal{L}(T)$ . Suppose furthermore that  $T|L_1$  and  $T|L_2$  do not consist of a single vertex. Then if  $T|L_1$  and  $T|L_2$  are edge-disjoint, they are also vertex-disjoint*

*Proof.* If two paths share an edge they also share the two end points of that edge, so the first claim follows.

For the second claim, suppose that  $T|L_1$  and  $T|L_2$  are edge disjoint but have the vertex  $v$  in common. If  $v$  is a leaf, then both  $T|L_1$  and  $T|L_2$  share the edge adjacent to that leaf - a contradiction.

If  $v$  is not a leaf, then there are  $u_1, v_1 \in L_1$  and  $u_2, v_2 \in L_2$  such that  $v \in \text{path}_T(u_1, v_1) \cap \text{path}_T(u_2, v_2)$ . But the degree of  $v$  is at most 3, therefore the two paths  $\text{path}_T(u_1, v_1)$  and  $\text{path}_T(u_2, v_2)$  have non-empty edge intersection - a contradiction. The proof follows. ■

Edge disjoint trees naturally define a forest.

**Lemma 3.5.** *Let  $T$  be a binary tree. Let  $L_1, \dots, L_\alpha$  be a partition of  $\mathcal{L}(T)$  and let  $(T_\gamma = T|L_\gamma)_{\gamma=1}^\alpha$  be a collection of (pairwise) edge disjoint trees. Then the tree  $T$  may be obtained from  $T_1, \dots, T_\alpha$  by adding  $\alpha - 1$  edges.*

*Proof.* Note first that if  $L = L_1 \cup L_2$  and  $T|L_1, T|L_2$  are edge disjoint then  $L_1|L_2 \in \Sigma(T)$ . Therefore, in this case,  $T$  may be obtained from  $T|L_1$  and  $T|L_2$  by adding a single edge.

In the general case, define  $\ell_T(T|L_\beta, T|L_\gamma)$  by

$$\min\{\ell_T(u, v) : u \in \text{path}_T(u', u''), v \in \text{path}_T(v', v''), u', u'' \in L_\beta, v', v'' \in L_\gamma\}.$$

Take  $\beta \neq \gamma$  that minimize the distance  $\ell(T|L_\beta, T|L_\gamma)$  among all pairs  $(\beta, \gamma)$ .

It is easy to see that  $T|L_\beta \cup L_\gamma$  is edge disjoint from  $T_{\beta'}$  for  $\beta' \notin \{\beta, \gamma\}$ . The general case follows by induction.

■

We say that an edge  $e \in \mathcal{E}(T)$  belongs to  $T|L$ , if there exist  $u, v \in L$  such that  $e \in \text{path}_T(u, v)$ . We say that the directed edge  $\vec{e}$  belongs to  $T|L$  if the edge  $e$  belongs to  $T|L$ . The distance of directed edge  $\vec{e} = \overrightarrow{(u_1, u_2)}$  to a set of vertices  $V' \subset \mathcal{V}(T)$  is the minimal length  $m - 1$  of a *simple* path  $(u_1, u_2), \dots, (u_{m-1}, u_m)$  such that  $u_m \in V'$ . We denote this distance by  $\bar{\ell}_T(\vec{e}, V')$ . Finally, let  $\ell_T^*(e, V') = \max(\bar{\ell}_T(\vec{e}, V'), \bar{\ell}_T(\overleftarrow{e}, V'))$  (the max is over the two orientations of  $e$ ).

**Lemma 3.6.** *Let  $T$  be a binary tree and  $L_1, \dots, L_\alpha$  be a partition of  $\mathcal{L}(T)$ . Suppose that  $(T|L_\beta)_{\beta=1}^\alpha$  is a collection of edge disjoint trees and that for all edges  $e \in \mathcal{E}(T)$  with  $\ell_T^*(e, \mathcal{L}(T)) \leq r$  the edge  $e$  belongs to one of the trees  $T|L_\beta$ . Then  $\alpha \leq 1 + 30 \times 2^{-r}n$ .*

Note that  $\ell_T^*(e = (u, v), V') \geq \min\{\ell(u, V'), \ell(v, V')\}$  and strict inequality may hold (see Figure 3). Thus the lemma *does not* follow from the fact that fractions of *vertices* at distance  $r$  from the set of leaves is at most  $2^{-r}$ .

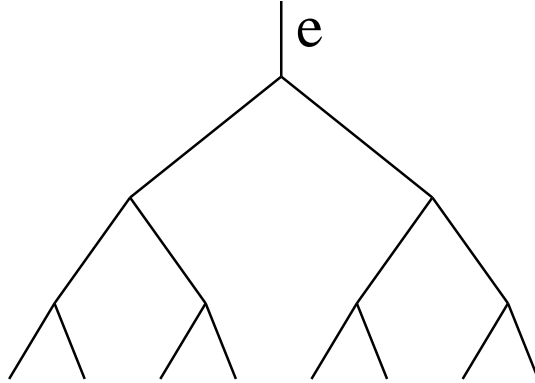


Figure 2: Note that  $\ell_T^*(e = (u, v), \mathcal{L}(T)) = 3$  while  $\min\{\ell(u, \mathcal{L}(T)), \ell(v, \mathcal{L}(T))\} = 0$

*Proof.* Following the argument of the previous lemma, it is easy to see that each edge we add must satisfy  $\ell_T^*(e, \mathcal{L}(T)) > r$ . Let  $A_m$  be the set of all edges whose  $\bar{\ell}$  distance to  $\mathcal{L}(T)$  is at least  $r$ . Then  $\alpha - 1 \leq |A_r|$ . It remains to bound the size of  $A_r$ .

Let  $b(n, m)$  be the maximal possible size of  $A_m$  among all binary trees on  $n$  leaves. Note that  $b(n, m) = 0$  if  $n \leq 2^{m-1}$ . Let  $T$  be a tree on  $n$  leaves and  $e$  an edge of  $T$ . Let  $T_1$  and  $T_2$  be the two trees obtained from  $T$  by removing the edge  $e$ . Note that the number of directed edges in  $T$  of  $\bar{\ell}$  distance at least  $m$  from  $\mathcal{L}(T)$  is at most five more than the sum of the number of such edges in  $T_1$  and  $T_2$  (we may add at most the new edge and the four new edges adjacent to it).

For every binary tree on  $n$  leaves there exists an edge  $e$  such that removing the edge  $e$  results in two trees  $T_1, T_2$  such that  $|\mathcal{L}(T_1)| \geq |\mathcal{L}(T)|/3$  and  $|\mathcal{L}(T_2)| \geq |\mathcal{L}(T)|/3$ . We therefore conclude that

$$b(n, m) \leq \max\{b(n_1, m) + b(n_2, m) + 5 : n_1 + n_2 = n \text{ and } n_1 \geq n/3 \text{ and } n_2 \geq n/3\}.$$

It now follows by easy induction that

$$b(n, m) \leq \max\{30 \times 2^{-m}n - 5, 0\}$$

for all  $n$ . In particular,

$$\alpha \leq 1 + |A_r| \leq 1 + \max\{30 \times 2^{-r}n - 5, 0\} \leq 1 + 30 \times 2^{-r}n.$$

as needed. ■

## 4 Super-trees for edge sharing trees

In this section we show how to build the super-tree of a collection of edge-sharing trees.

**Definition 4.1.** Let  $T$  be a binary tree and  $L_1, L_2, \dots, L_\alpha \subset \mathcal{L}(T)$ . We say that  $T|L_1, \dots, T|L_\alpha$  are edge sharing if there is no partition  $S_1 \cup S_2$  of  $\{1, \dots, \alpha\}$  such that  $T|L_\beta$  and  $T|L_\gamma$  are edge disjoint for all  $\beta \in S_1$  and  $\gamma \in S_2$ .

From Lemma 3.3 it follows that

**Proposition 4.1.** Let  $T$  be a binary tree and  $L_1, L_2, \dots, L_\alpha \subset \mathcal{L}(T)$ . Then  $T|L_1, \dots, T|L_\alpha$  are edge sharing if and only if there is no partition  $S_1 \cup S_2$  of  $\{1, \dots, \alpha\}$  for which  $T|\cup_{\beta \in S_1} L_\beta$  and  $T|\cup_{\beta \in S_2} L_\beta$  are edge disjoint.

In the main result of this section we prove the following

**Theorem 4.2.** Let  $T$  be a binary tree and  $L_1, L_2, \dots, L_\alpha \subset \mathcal{L}(T)$  such that  $T|L_1, \dots, T|L_\alpha$  are edge sharing. Let  $L' = \cup_{\beta=1}^\alpha L_\beta$  and  $T' = T|L'$ . For  $1 \leq \beta \leq \alpha$ , let

$$S(\beta) = \{\gamma : T|L_\beta \text{ and } T|L_\gamma \text{ are edge sharing}\},$$

and let  $SL_\beta = \cup_{\gamma \in S(\beta)} L_\gamma$ . Then

- The tree  $T'$  is determined by the trees  $(T|SL_\beta)_{\beta=1}^\alpha$ .
- Moreover, given the trees  $(T|SL_\beta)_{\beta=1}^\alpha$ , there is a polynomial time algorithm that computes  $T'$ .

Theorem 4.2 states that it is possible to glue together a collection of edge-sharing trees, given some “local” tree structures.

**Lemma 4.3.** Assume the setting of Theorem 4.2. Let  $e \in \mathcal{E}(T|SL_\beta)$  satisfy that there exist  $u, v \in L_\beta$  with  $e \in \text{path}_{T|SL_\beta}(u, v)$ . Let  $A_0|B_0$  be the partition defined by  $e$  on  $SL_\beta$ . Then there exists a unique partition  $A|B \in \Sigma(T')$  such that  $A_0 = A \cap SL_\beta$  and  $B_0 = B \cap SL_\beta$ .

Moreover, for every edge  $\tilde{e} \in \mathcal{E}(T')$ , there exists  $1 \leq \beta \leq \alpha$ , leaves  $u, v \in L_\beta$  and  $e \in \text{path}_{T|SL_\beta}(u, v)$  such that the partition of  $T|SL_\beta$  defined by  $e$  is given by  $A \cap SL_\beta|B \cap SL_\beta$ , where  $A|B$  is the partition defined by  $\tilde{e}$ .

*Proof.* Since  $T'|SL_\beta = T|SL_\beta$ , there exists a partition  $A|B$  of  $L'$  corresponding to an edge of  $T'$  which satisfies  $A \cap SL_\beta = A_0$  and  $B \cap SL_\beta = B_0$ . Thus, in order to prove the first claim, it remains to show that  $A|B$  is unique.

Write  $e = (x_1, x_{k+1})$  and let  $(x_1, x_2), \dots, (x_k, x_{k+1})$  be the path corresponding to  $e$  in  $T'$ . Note that  $A|B$  defines a partition satisfying  $A_0 = A \cap SL_\beta$  and  $B_0 = B \cap SL_\beta$  if and only if  $A|B$  corresponds to one of the edges  $(x_m, x_{m+1})$ . The last claim follows from the fact that removing an edge of  $T'$  that doesn't correspond to any edge in  $T|SL_\beta$  induces the trivial partition on  $SL_\beta$  and removing an edge  $\tilde{e}$  that correspond to the edge  $e$  of  $T|SL_\beta$  corresponds to the partition defined by  $e$ .

Therefore, it suffices to show that  $k = 1$ .

Suppose that  $k > 1$ . Since the edge  $(x_1, x_2)$  is not defined in  $T'|SL_\beta$ , and the collection  $(T|S_\beta)_{\beta=1}^\alpha$  is edge-sharing, it follows that there exists  $(u, v) \in L_\gamma$  such that  $(x_1, x_2) \in \text{path}(u, v)$  and  $\gamma \notin S(\beta)$ . But the fact that  $(x_1, x_2) \in \text{path}(u, v)$  implies that  $T|L_\beta$  and  $T|L_\gamma$  are edge sharing - a contradiction. Therefore  $k = 1$  and the first claim follows.

For the second claim, note that Lemma 3.2 and Proposition 4.1 imply that for all  $\tilde{e} \in \mathcal{E}(T')$ , there exists  $1 \leq \beta \leq \alpha$  and  $u, v \in L_\beta$  such that  $\tilde{e} \in \text{path}_{T'}(u, v)$ . By the previous argument, the edge  $\tilde{e}$  corresponds to a unique edge  $e \in \mathcal{E}(T|SL_\beta)$ , as needed. ■

**Proof Of Theorem 4.2:** We will show how to reconstruct  $T'$  in time polynomial in  $n$ . Clearly, it suffices to show how to reconstruct  $\Sigma(T')$  in polynomial time. From Lemma 4.3 it follows that in order to find  $\Sigma(T')$  it suffices to find for all  $1 \leq \beta \leq \alpha$  and all edges  $e \in \mathcal{E}(T|SL_\beta)$  which satisfy  $e \in \text{path}_{SL_\beta}(u, v)$  for  $u, v \in L_\beta$ :

- The partition  $A_0|B_0$  of  $SL_\beta$  corresponding to the edge  $e$ .
- The unique partition  $A|B$  of  $L'$  satisfying  $A_0 = A \cap SL_\beta$  and  $B_0 = B \cap SL_\beta$ .

Given  $T|SL_\beta$  it is trivial to find the partition  $A_0|B_0$  of  $SL_\beta$  corresponding to  $e$ . All that remains to show is how to find the partition  $A|B$  corresponding to the edge  $e$  in  $T'$ . This is the unique partition satisfying  $A_0 = A \cap L'$  and  $B_0 = B \cap L'$ .

For  $S' \supset S(\beta)$  let  $\tilde{L} = \cup_{\gamma \in S'} L_\gamma$ . Lemma 4.3 allows to identify the edge  $e$  in  $T|SL_\beta$  with the unique edge corresponding to  $e$  in  $T|\tilde{L}$ . We will use this identification below.

We now give an inductive construction of  $A$  and  $B$ . Let  $S_0(\beta) = S(\beta)$  and continue inductively by letting for  $d \geq 1$

- $S_d(\beta) = \{\gamma : \exists \delta \in S_{d-1}(\beta) \text{ s.t. } (L_\delta, L_\gamma) \text{ are edge sharing}\} \setminus (\cup_{c < d} S_c(\beta))$ .
- Let  $A_d = A_{d-1} \cup_{\gamma \in S_d^A(\beta)} L_\gamma$ , where  $\gamma \in S_d(\beta)$  belongs to  $S_d^A(\beta)$  if the following condition holds: There exists leaves  $u', v' \in A_{d-1}$  and leaves  $u, v \in L_\gamma$  such that  $\text{path}(u', v') \cap \text{path}(u, v) \neq \emptyset$ .
- Similarly, let  $B_d = B_{d-1} \cup_{\gamma \in S_d^B(\beta)} L_\gamma$ , where  $\gamma \in S_d(\beta)$  belongs to  $S_d^B(\beta)$  if the following condition holds: There exists leaves  $u', v' \in B_{d-1}$ , leaves  $u, v \in L_\gamma$  such that  $\text{path}(u', v') \cap \text{path}(u, v) \neq \emptyset$ .

The above construction is repeated until  $S_d(\beta) = \emptyset$ .

We now prove the validity of the construction. First, from the fact that  $T|L_1, \dots, T|L_\alpha$  are edge sharing, it follows that  $\{1, 2, \dots, \alpha\} = \cup_{d \geq 0} S_d(\beta)$ . We write  $\bar{S}_d(\beta)$  for  $\cup_{c \leq d} S_c(\beta)$ .

**Claim 4.4.** For all  $d \geq 0$  it holds that  $S_d(\beta) = S_d^A(\beta) \cup S_d^B(\beta)$ . For all  $d \geq 0$ , the partition  $A_d|B_d$  is the partition of the tree  $T|_{\cup_{\gamma \in \bar{S}_d(\beta)} L_\gamma}$  defined by the edge  $e$ .

*Proof.* The proof is by induction on  $d$ . The base case  $d = 0$  is immediate. For the inductive step note that under the induction hypothesis for  $d - 1$ , the partition  $A_{d-1}|B_{d-1}$  is the partition induced by  $e$  on the tree  $T|_{\cup_{\gamma \in \bar{S}_{d-1}(\beta)} L_\gamma}$ .

Let  $\gamma \in S_d(\beta)$ . Clearly  $T|_{L_\gamma}$  and  $T|_{A_{d-1} \cup B_{d-1}}$  share edges. On the other hand since  $e$  is not an edge of  $T|_{L_\gamma}$  it follows that either  $T|_{L_\gamma}$  and  $T|_{A_{d-1}}$  share edges or  $T|_{L_\gamma}$  and  $T|_{B_{d-1}}$  share edges. In the first case  $L_\gamma \subset A_d$ , while in the second case  $L_\gamma \subset B_d$ .

The claim follows. ■

The proof of the theorem follows as when the algorithm terminates, the sets  $A_d|B_d$  define the desired partition. It is clear that the algorithm described above runs in polynomial time. ■

## 5 Distorted metrics on trees

In this section we will prove Theorem 2.1. We will assume (2) below (i.e.  $f \leq d(e) \leq g$  for all  $e \in E$ ).

It is helpful to define “balls” with respect to  $d$  and  $\hat{d}$  as follows.

$$B_L(v, r) = \{w \in \mathcal{L}(T) : d(v, w) \leq r\}, B_V(v, r) = \{w \in \mathcal{V}(T) : d(v, w) \leq r\}. \quad (4)$$

We similarly define  $\hat{B}_L(v, r)$  and  $\hat{B}_V(v, r)$  with  $\hat{d}$  instead of  $d$ .

We omit the proof of the following easy Lemma.

**Lemma 5.1.** Let  $\hat{d}$  be an  $(\epsilon, M)$  distortion of  $d$  and let  $r < M$ . Consider the sets  $L_\beta = \hat{B}_L(v_\beta, r)$  for  $\beta = 0, \dots, \alpha$ . Suppose that  $T|_{L_0}$  and  $T|_{L_\beta}$  are edge sharing for  $\beta = 1, \dots, \alpha$ . Then for all  $u, u' \in \mathcal{V}(T|_{\cup_{\beta=0}^\alpha L_\alpha})$  it holds that

$$d(u, v) < 6r + 6\epsilon. \quad (5)$$

Similarly, for all  $u, u' \in \cup_{\beta=0}^\alpha L_\alpha$  it holds that

$$\hat{d}(u, v) < 6r + 7\epsilon, \quad (6)$$

if  $\hat{d}(u, v) < \infty$ .

*Proof.* Equation (6) follows immediately from (5). Let  $T' = T|_{\cup_{\beta=0}^\alpha L_\beta}$ . Note that if  $\beta \leq \alpha$ , the leaves  $u_1, u_2$  belong to  $L_\beta$  and  $u$  is a vertex on the path connecting  $u_1$  and  $u_2$ , then

$$d(u, v_\beta) \leq \sup_{w \in L_\beta} d(w, v_\beta) < \sup_{w \in L_\beta} \hat{d}(w, v_\beta) + \epsilon \leq r + \epsilon. \quad (7)$$

(the first inequality follows from the fact that if  $v$  is a vertex in a tree with a path metric, then the vertex furthest away from  $v$  is a leaf).

Now let  $u, u' \in \mathcal{V}(T')$ . Since the trees  $(T|_{L_\beta})_{\beta \leq \alpha}$  are edge sharing, it follows  $u$  belongs to a path connecting two points in  $L_\gamma$  and  $u'$  belongs to a path connecting two point in  $L_{\gamma'}$ , where  $0 \leq \gamma, \gamma' \leq \beta$ .

Let  $w$  be a vertex that belongs to an edge  $e$  such that  $e \in \text{path}(w_{\gamma,1}, w_{\gamma,2}) \cap \text{path}(w_{0,1}, w_{0,2})$  where  $w_{\gamma,1}, w_{\gamma,2} \in L_\gamma$  and  $w_{0,1}, w_{0,2} \in L_0$ . Define  $w'$  similarly. Then by (7)

$$\begin{aligned} d(u, u') &\leq d(u, v_\gamma) + d(v_\gamma, w) + d(w, v_0) + d(v_0, w') + d(w', v_{\gamma'}) + d(v_{\gamma'}, u') \\ &\leq 6(r + \epsilon), \end{aligned}$$

as needed. ■

**Proof of Theorem 2.1:** Let

$$r = \frac{M-7\epsilon}{6}, \quad L = (v_\beta)_{\beta=1}^n, \quad L_\beta = \hat{B}(v_\beta, r).$$

Define a graph  $G$  on the set of vertices  $\{1, \dots, n\}$  where the edge  $(\beta, \gamma)$  is present if and only if  $L_\beta$  and  $L_\gamma$  are edge-sharing. Let  $C_1, \dots, C_\alpha$  be the partition of  $\{1, \dots, n\}$  to connected components in  $G$ . It is easy to see that the graph  $G$  can be computed in polynomial time. Indeed by Lemma 5.1 it follows that if  $L_\beta$  and  $L_\gamma$  share edges then for all  $u \in L_\beta$  and  $v \in L_\gamma$  it holds that  $\hat{d}(u, v) \leq M$ . For sets  $L_\beta, L_\gamma$  for which this condition holds, we may easily reconstruct the tree  $T|_{L_\beta \cup L_\gamma}$  using the 4-point method. We can then check if  $T|_{L_\beta}$  and  $T|_{L_\gamma}$  are edge sharing in  $T|_{L_\beta \cup L_\gamma}$ .

By proposition (4.1) it follows that if  $\sigma \neq \tau$  then the trees  $T|_{\cup_{\beta \in C_\sigma} L_\beta}$  and  $T|_{\cup_{\beta \in C_\tau} L_\beta}$  are edge disjoint. Moreover, if  $1 \leq \eta \leq \alpha$ , then the collection of trees  $(T|_{L_\beta})_{\beta \in C_\eta}$  are edge sharing.

Note that in the notation of Theorem 4.2 for all  $\beta$  and all  $u, v \in SL_\beta$  it holds that  $\hat{d}(u, v) \leq M$  - by Lemma 5.1.

It follows that the trees  $T|_{SL_\beta}$  may be easily recovered by the 4-point method. Moreover, for every edge  $e \in \mathcal{E}(T|_{SL_\beta})$  we may recover  $\hat{d}(e)$  satisfying  $|d(e) - \hat{d}(e)| < 2\epsilon$ .

We now use Theorem 4.2 in order to recover the trees  $T|_{\cup_{\beta \in C_\sigma} L_\beta}$  for all  $\sigma$ . Moreover, we may recover  $\hat{d}(e)$  satisfying  $|d(e) - \hat{d}(e)| < 2\epsilon$ .

It remains to bound the number of trees  $\alpha$  using Lemma 3.6. Note that if  $l_T^*((u, v), \mathcal{L}(T)) \leq p$ , then there is a path of  $p-1$  edges starting at  $u$ , avoiding  $v$  and ending at  $\mathcal{L}(T)$  at a node denoted  $u'$ . Similarly, there is a path of  $p-1$  edges starting at  $v$  avoiding  $u$  and ending at  $\mathcal{L}(T)$  at a node denoted  $v'$ . Note that  $d(u', v') \leq (2p-1)g$  and therefore  $\hat{d}(u', v') \leq (2p-1)g + \epsilon$ . Thus if  $p \leq \frac{1}{2} + \frac{M-\epsilon}{2g}$ , then  $\hat{d}(u', v') \leq M$  and therefore the edge  $e$  belongs to one of the trees  $T|_{\cup_{\beta \in C_\sigma} L_\beta}$ . It follows from Lemma 3.6 that

$$\alpha \leq 1 + 30n \times 2^{-\lfloor \frac{1}{2} + \frac{M-\epsilon}{2g} \rfloor} \leq 1 + \frac{60n}{\sqrt{2}} \times 2^{-\frac{M-\epsilon}{2g}}.$$

The theorem follows. ■

## References

- [1] P. Buneman. The recovery of trees from measures of dissimilarity. In *Mathematics in the Archaeological and Historical Sciences*, pages 187–395. Edinburgh University Press, Edinburgh, 1971.
- [2] J. A. Cavender. Taxonomy with confidence. *Math. Biosci.*, 40(3-4), 1978.

- [3] M. Cryan, L. A. Goldverg, and P. W. Goldberg. Evolutionary trees can be learned in polynomial time in the two-state general markov mode. *SIAM Journal on Computing*, 31(2):375–397, 2003. The short version appeared in Proceedings of the 39th Annual Symposium on Foundations of Computer Science, 1998.
- [4] W. Day, D. Johnson, and D. Sankoff. The computational complexity of inferring rooted phylogenies by parsimony. *Mathematical biosciences*, 86:33–42, 1986.
- [5] P. L. Erdős, M. A. Steel, L. A. Székely, and T. A. Warnow. A few logs suffice to build (almost) all trees (part 1). *Random Structures Algorithms*, 14(2):153–184, 1999.
- [6] P. L. Erdős, M. A. Steel, L. A. Székely, and T. A. Warnow. A few logs suffice to build (almost) all trees (part 2). *Theoretical Computer Science*, 221:77–118, 1999.
- [7] M. Farach and S. Kannan. Efficient algorithms for inverting evolution. *J. ACM.*, 46(4):437–449, 1999.
- [8] J. Felsenstein. Cases in which parsimony or compatibility methods will be positively misleading. *Syst. Biol.*, pages 401–410, 1978.
- [9] J. Felsenstein. *Inferring Phylogenies*. Sinauer, New York, New York, 2004.
- [10] L. R. Foulds and R. L. Graham. The steiner problem in phylogeny is npcomplete. *Adv. Appl. Math*, 3:43–49, 1982.
- [11] R. L. Graham. and L. R. Foulds. Unlikelihood that minimal phylogenies for a realistic biological study can be constructed in reasonable computational time. *Math. Biosci.*, 60:133–142, 1982.
- [12] D. Gusfield. Efficient algorithms for inferring evolutionary trees. *Networks*, 21, 1991.
- [13] E. Mossel. On the impossibility of reconstructing ancestral data and phylogenies. *Jour. Comput. Bio.*, 10(5):669–678, 2003.
- [14] E. Mossel. Phase transitions in phylogeny. To appear in Trans. of AMS., 2003.
- [15] E. Mossel and M. Steel. A phase transition for a random cluster model on phylogenetic trees. Submitted to Mathematical Biosciences, 2003.
- [16] C. Semple and M. Steel. *Phylogenetics*, volume 22 of *Mathematics and its Applications series*. Oxford University Press, 2003.
- [17] E. Sober and M. A. Steel. Testing the hypothesis of common ancestry. *Jour. Theor. Bio.*, 218:395–408, 2002.
- [18] M. Steel. Recovering a tree from the leaf colourations it generates under a Markov model. *Appl. Math. Lett.*, 7(2):19–23, 1994.
- [19] M. A. Steel and L. A. Székely. Inverting random functions. II. Explicit bounds for discrete maximum likelihood estimation, with applications. *SIAM J. Discrete Math.*, 15(4):562–575 (electronic), 2002.



## A Large deviations for the log-det distance

In this section, we prove Proposition 2.2. We will use the following large deviation result from [6].

**Lemma A.1 ([6]).** *For every two vertices  $u, v \in \mathcal{L}(T)$  and  $a, b \in \mathcal{A}$ , let  $F_{a,b}$  be the probability that node  $u$  has letter  $a$  and node  $w$  has letter  $b$ . Let  $-d(u, v) = \log \det(F_{i,j})$ . Then  $d(u, v)$  is a path metric on the tree satisfying*

- $d(e) \geq f$  for all edges  $e$  of the tree.
- For all  $u, v \in \mathcal{L}(T)$ :

$$\begin{aligned} d(u, v) &\leq -|\mathcal{A}| \log \pi_{\min} - \sum_{e \in \text{path}(u, v)} \log \det(M(e)) \\ &\leq -|A| \log \pi_{\min} + 2g|\text{path}(u, v)|. \end{aligned}$$

Moreover for  $u, v \in \mathcal{L}(T)$  let  $\hat{F}_{a,b}$  be the empirical distribution of having  $a$  at  $u$  and  $v$  in  $b$  in a collection of  $k$  samples. Let  $\hat{d}(u, v) = -\log \det(\hat{F}_{a,b})$  if  $\det(\hat{F}_{a,b}) > 0$  and  $\hat{d}(u, v) = \infty$  otherwise. Then there exists positive constants  $c_1$  and  $c_2$  such that

$$\mathbf{P}[|e^{-d(u,v)} - e^{-\hat{d}(u,v)}| \geq t] \leq 2 \exp\left(-c_1 k \left(t - \frac{c_2}{k}\right)_+^2\right), \quad (8)$$

where  $(a)_+ = \max\{0, a\}$ .

For the proof see [6, Section 7]. Equation (8) here is equation (49) in [6] up to change of notation. Taking  $t = e^{-M-\epsilon}(1 - e^{-2\epsilon})$  in (8) we see that if  $|e^{-d(u,v)} - e^{-\hat{d}(u,v)}| \leq t$  and either  $d(u, v) \leq M + \epsilon$  or  $\hat{d}(u, v) \leq M + \epsilon$ , then  $|d(u, v) - \hat{d}(u, v)| < \epsilon$ . So if  $|e^{-d(u,v)} - e^{-\hat{d}(u,v)}| \leq t$  for all  $u$  and  $v$ , then  $\hat{d}$  is an  $(\epsilon, M)$  distortion of  $d$ .

Taking  $k$  that satisfies (3) we obtain that the error is at most

$$n^2 \exp\left(-c_1 k \left(t - \frac{c_2}{k}\right)_+^2\right) \leq n^{2-r},$$

if  $c$  (in (3)) is sufficiently large. The proof of Proposition 2.2 follows.

## B Lower bounds

In this section we prove tightness of both the distorted metric result and the phylogenetic reconstruction result.

The tightness of the metric result follow easily by considering the  $r$ -level 3-regular tree with the metric  $d$  that assigns length  $d$  to all edges of the tree. We let  $\hat{d}(u, v) = d(u, v)$  if  $d(u, v) \leq M$  and  $\hat{d}(u, v) = \infty$  otherwise. Then  $\hat{d}$  is a  $(0, \infty)$  distortion of  $d$ .

Define the relation  $u \sim v$  if  $d(u, v) \leq M$ . It is easy to see that  $\sim$  is an equivalence relation. There are  $n2^{-\lfloor M/2g \rfloor}$  equivalence classes for this relation. It is easy to reconstruct the tree on each class, but since for  $u, v$  which belong to different classes,  $\hat{d}(u, v) = \infty$ , it is impossible to reconstruct

any more. This prove the tightness of the number of trees in Theorem 2.1 up to a multiplicative constant.

A similar construction yields the analogous sampling complexity lower bound for phylogenetic trees. We fix the model to be the CFN model where the length of each edge is  $g$ . Thus the mutation matrices are given by  $M(e) = \begin{pmatrix} e^{-g} + (1 - e^{-g})/2 & (1 - e^{-g})/2 \\ (1 - e^{-g})/2 & e^{-g} + (1 - e^{-g})/2 \end{pmatrix}$ .

Thus for each edge  $(u, v)$ , the state of  $u$  is copied to  $v$  with probability  $e^{-g}$ . Otherwise, an independent uniform state is chosen.

Following the arguments of [13] implies that if  $v$  is a vertex at  $\ell$ -distance  $s$  from the set of leaves, then the character value at the leaves below  $v$  is independent of the character at  $v$  with probability at least  $1 - 2^s e^{-gs}$ . Thus the character at the leaves is independent from all nodes at level  $s$  with probability at least  $1 - 3 * 2^{r-s-1} 2^2 e^{-gs} = 1 - ne^{-gs}$ . The probability that the former event will occur for  $k$  characters is at least  $p_k = 1 - kne^{-gs}$ .

Let assume further that the phylogenetic tree on each of the equivalence classes of the relation  $\sim$  defined by  $u \sim v$  if  $d(u, v) \leq 2gs$  is given. Then with probability  $p_k$ , there is no non-trivial information about the ancestral relationship except that given by the given  $n * 2^{-s}$  trees.

Note furthermore that  $p_k \geq 1 - \delta$  if  $k \leq \delta e^{gs}/n = \delta e^M/n$ . This proves the tightness of condition (3) in Theorem 2.3 up to a factor 2 in the exponent and a multiplicative  $O(n)$  factor.

## C Variants of the method

We briefly sketch a few variants of the method which may be practical advantages over the method analyzed here.

### C.1 Checking if two balls define tree disjoint trees

The first stage of the algorithm consists of checking if two balls  $\hat{B}_L(v, r)$  and  $\hat{B}_L(u, r)$  define two edge-sharing trees or two edge-disjoint trees. Most of the work at this stage is devoted to couples of trees that are edge-disjoint. In fact for most such pairs it would hold that  $\hat{d}(u, v) > M$  which implies automatically edge-disjointness without additional computation. Thus the efficient way of computing the graph  $G$  is by first checking for each  $u$  and  $v$  if  $\hat{d}(u, v) > M$ . Otherwise, we perform the test described in the proof of Theorem 2.1.

### C.2 Building supertrees for edge disjoint trees

A lot of computational effort is devoted to building super-trees from collection of edge-sharing trees. There are many variants that work here. Instead of the method described in the paper, we can use quartets method as in [5]. Similarly to [5] one can prove that given a collection of edge sharing trees  $T_1, \dots, T_\alpha$ , their super-tree is in fact defined by all quartets belonging to the trees  $T|SL_\beta$  for  $1 \leq \beta \leq \alpha$  via the dyadic closure operator. This may lead to a computationally more efficient algorithm than ours.