Building Consensus with Balanced Splits *

Avrim Blum[1], Jamie Morgenstern[2], R. Ravi[3], and Santosh Vempala[4]

[1] Computer Science Department, Carnegie Mellon University, USA, avrim@cs.cmu.edu.
[2] Computer Science Department, Carnegie Mellon University, USA, jamiemmt@cs.cmu.edu
[3] Tepper School of Business, Carnegie Mellon University, USA, ravi@cmu.edu.
[4] College of Computing, Georgia Institute of Technology, USA, vempala@cc.gatech.edu

Abstract. Given the multitude of sources for reconstructing the evolutionary history between entities, phylogenetic reconstruction methods often provide several trees classifying these entities. A key step in reconciling the different trees is to construct a consensus view of the history. If we consider each tree as a collection of laminar splits (two-way partitions), the primary problem in arriving at a consensus is to find a single split that agrees best with the closest split in each of the trees. To prevent non-informative splits, we should eliminate splits that are not balanced. We formulate this balanced consensus cut problem as one of maximizing the correlation (number of agreements minus disagreements), and provide a .611-approximation based on rounding an SDP formulation. The SDP formulation uses new sets of product variables that may be useful in other contexts.
We also consider similar consensus problems formulated as minimizing(and maximizing) the amount of disagreement(agreement) between a final split and an input collection of splits, trees or chains (permutations). For these cases, our results include polynomial-time exact algorithms as well as PTAS's and quasi-PTASs based on sampling and dynamic programming. Our work opens the door to many new and interesting problems in finding consensus splits to reconcile various input trees in phylogenetic reconstruction.

Keywords: Approximation Algorithms, Phylogenetic Trees, Correlation and Consensus, Clustering, Stars, Metrics, SDP, Integrality Gap.

1 Introduction

The reconstruction of evolutionary history from natural records is a rich area of study, with a wide variety of methods designed to uncover hidden structure in the discrete character variations among the classified entities (such as the presence/absence of a vertebral column) or in distances between genetic sequences coding for particular genes (such as the mutation distances between copies of the Haemoglobin gene among the various species). Both character- and distance-based methods have been developed to construct rooted binary tree topologies with the nodes labeled with the entities (species).

Reconciling phylogenetic trees. The explosive availability of genetic data from the various genome projects and cheap sequencing methods have allowed the generation of several possible phylogenetic trees among the same set of species by examining different portions of their genetic sequences. In another case, by examining the variations among various sub-populations within the same species (such as the human races), and building an evolutionary tree from the changes between the groups, one can construct multiple possible trees explaining recent human evolution, and the associated questions of human migration. In Sridhar et al. [2007a,b], by using short windows of genetic sequences over the publicly available HapMap (of varying haplotypes among human sub-populations), tens of thousands of trees are derived between the various racial groups. A key question that arises is to find a consensus tree from all these input trees such that the splits represented by this tree has a high confidence of correctness Tsai et al. [2011].

* This work was supported in part by NSF grants CCF-1116892, IIS-1065251, and CCF-1218382.

In this paper we focus on a core algorithmic challenge in reconciling phylogenetic trees: the problem of finding a balanced consensus cut. If we view each input tree as a collection of laminar splits induced by the edges in the tree, the goal in the *balanced consensus cut* is to find a single balanced cut that as much as possible agrees with some split within each input tree (the split induced by some edge in each tree). The reason for requiring a balance condition is that otherwise a trivial but uninformative solution is to just separate one of the input species from all the others, which would agree with all input trees but not identify any common structure that they have. We also consider a number of extensions of this basic problem.

Consensus Meta-feature. The problem of finding a balanced consensus cut also arises naturally in a variety of other settings. Consider the case of reconciling the results from multiple surveys where each survey/interviewer asks several questions of the same population: each question splits the subject population. Thus each interviewer defines a set of splits of the population, and wish to find a feature (split) that is most consistent with the input of all the interviewers. We can formulate this as finding a balanced cut that agrees with at least one question from each survey. Note that here the set of splits corresponding to each interviewer may not be a laminar family, but in fact our algorithms will apply to this generalization as well.

Problems studied. Even in the case that each input (each laminar family of splits) has just a single split, the problem of finding the cut of maximum agreement is NP-hard [Dörnfelder et al., 2011]. Therefore, we focus primarily on approximations. We have three natural choices in defining the objective we are approximating: we can maximize the total agreement (where for each input we choose the closest split to our cut, and then sum over all inputs), we can minimize the total amount of disagreement, or we can look at correlation: the total agreement minus the total disagreement. We consider all three types of objectives in the sequel. One of our main results is a new SDP-rounding based approximation algorithm for the correlation objective. We also devise QPTAS's for the agreement and disagreement versions using sampling and dynamic programming. In the Appendix, we also consider other restricted forms of inputs such as chain families (laminar families representing permutations) and also other restricted forms of labeled trees as inputs where we show we can reduce the problem to a min-cost matching and therefore efficiently compute exact optimal solutions. We also consider more general types of outputs such as trees rather than cuts.

Balanced Cut Maximizing Agreement

Define a *split* $(S, \bar{S})$ as a partition of $[n]$ into a subset S and its complement $\bar{S}$. A split is unordered, i.e., $(S, \bar{S}) = (\bar{S}, S)$. The input to the **Maximum Balanced Cut** (MaxBC) problem is a set of m families of splits, $\mathrm{L}_1, \ldots, \mathrm{L}_m$, each family consisting of a set of N splits. (E.g., in the case of trees, each L_i would be a laminar family and N would be $n-1$.) The output is a cut, i.e., a subset X of $[n]$. Viewing X and S as ± 1 indicator vectors, the *agreement* of a split $(S, \bar{S})$ of $[n]$ with the cut X is $A(S, X) = \max\{\sum_j \mathbf{1}_{X_j = S_j}, \sum_j \mathbf{1}_{X_j \neq S_j}\}$. The goal is to find a consensus cut of size k (given) and maximum total agreement. This leads to the following maximization agreement objective.

$$f(X) = \sum_{i=1}^{m} \max_{S \in \mathrm{L}_i} A(S, X).$$

In words, we want to maximize the sum of agreements with the closest match within each family L_i. Note that *any* cut X achieves at least half of optimal, so the goal is to perform significantly better. For this objective, we get a QPTAS, which becomes a PTAS when $N = O(1)$.

Balanced Cut Minimizing Disagreement

In the **Minimum Balanced Cut** (MinBC) problem, again our input is a set of m families of splits, $\mathrm{L}_1, \ldots, \mathrm{L}_m$, each family consisting of a set of N splits. Viewing X and S as ± 1 indicator vectors, the *disagreement* of a split $(S, \bar{S})$ of $[n]$ with the cut X is

$$D(S, X) = \min(\sum_j \mathbf{1}_{X_j \neq S_j}, \sum_j \mathbf{1}_{X_j = S_j}) = \frac{1}{2} \min(\|S - X\|_1, \|S - \bar{X}\|_1).$$

The goal now is to find a consensus cut of size k (given) and minimum total disagreement. This leads to the following minimization agreement objective:

$$g(X) = \sum_{i=1}^{m} \min_{S \in \mathrm{L}_i} D(S, X).$$

Note that this problem (and more general variants of it detailed in the Appendix) admits a simple 2-approximation algorithm: simply use the best cut available from one of the input collections as the output solution. Suppose the optimal solution is X^* and the minimum S in L_i achieving the minimum in the objective is the split S_i. Furthermore, assume again wlog that $D(S_1, X^*) \leq D(S_i, X^*)$ for all $i \geq 2$. Note that one of the choices considered by our simple algorithm is to make S_1 the output cut. For this choice, the cost of the solution is

$$\sum_{i=1}^{m} \min_{S \in \mathrm{L}_i} D(S, S_1) \leq \sum_{i=1}^{m} D(S_i, S_1) \leq \sum_{i=1}^{m} \frac{1}{2}\|S_i - X^*\|_1 + \frac{1}{2}\|X^* - S_1\|_1 \leq \sum_{i=1}^{m} 2D(S_i, X^*).$$

The second inequality above is from triangle inequality and the last quantity is twice the optimal disagreement by definition. This argument is simply an instantiation of a more general such argument in minimization problems with metric costs [Wong, 1980]. Our main result on MinBC is a substantial strengthening of this result using sampling and dynamic programming; the same algorithm works for MaxBC as well.

Theorem 1. *For any $\varepsilon < 1/2$. we can get a $(1+\varepsilon)$-relative approximation for the MinBC and a $(1-\epsilon)$-relative approximation for the MaxBC problem in time $m \cdot n^{O(\frac{\log(N+1)}{\varepsilon^2})}$ if each input family has at most N splits. Note that for $N = O(1)$ this is a PTAS.*

Balanced Cut Maximizing Correlation

In the **Maximum Correlation Balanced Consensus Cut** (MaxBCorr) problem, we aim to maximize correlation, that is, agreements minus disagreements. Since we will be concerned only with cuts $X \subset [n]$ of size exactly $n/2$, we can equivalently define the correlation of a split $(S, \bar{S})$ with X, viewing X as a ± 1 indicator vector, as:

$$C(S, X) = \max\{\sum_{j \in S} X_j, \sum_{j \in \bar{S}} X_j\}$$

In particular, this quantity equals $\max(|S \cap X| - |S \cap \bar{X}|, |\bar{S} \cap X| - |\bar{S} \cap \bar{X}| = \frac{1}{2}(|S \cap X| + |\bar{S} \cap \bar{X}| - (|S \cap \bar{X}| + |\bar{S} \cap \bar{X}|)) = \frac{1}{2}(\text{agreements} - \text{disagreements})$. The goal now is to find a consensus cut of size $n/2$ to maximize

$$h(X) = \sum_{i=1}^{m} \max_{S \in \mathrm{L}_i} C(S, X).$$

Our main result is an approximation using a semi-definite programming (SDP) relaxation of an integer quadratic programming formulation of the problem.

Theorem 2. *For every instance of the MaxBCorr problem with $k = \frac{n}{2}$, there is an approximation algorithm that returns a bisection with expected correlation objective at least .611 times that of the optimal.*

1.1 Summary of Results

Our results, in summary, are as follows:

1. .611-approximation for MaxBCorr, Section 3
2. QPTAS for MaxBC, input families size $\leq N$, $n^{O(\log(N)/\epsilon^2)}$, Section 4
3. QPTAS for MinBC, input families size $\leq N$, $n^{O(\log(N)/\epsilon^2)}$, Section 4 (PTAS for $N = O(1)$)

4. 2-approximation for MinBC, Section 1
5. Exact solution for aligned trees to trees, Section 5
6. Almost-aligned trees (with no degree-2 nodes) of depth $\leq d$ to aligned trees, QPTAS, $O(n^{(d^2/\epsilon^2)})$, Section A
7. Exact solution for Chains to permutation, Section B

where a **tree** is a rooted laminar family, an **aligned** tree is one in which each cut associated with an edge in that tree is labeled left or right (and comparing two trees will be done by comparing, element by element, the distance between their labelings according to these L/R cuts), and an **almost-aligned** tree has each cut specified with a L and R side, except for the cut at the root, for which L and R can be selected to minimize cost. We present the proofs of Theorem 1, Theorem 2, and the exact resut for aligned trees, in the body of the paper and defer a formal description of our other results to the Appendix.

2 Related Work

Correlation clustering Bansal et al. [2002] introduced the *correlation clustering* problem: given a collection of n vertices, and a graph G where each edge is labeled $+1/-1$, compute a clustering which minimizes the number of $+$ edges cut by the clustering and the number of $-$ edges within a cluster (or maximizes the number of $+$ edges within a cluster plus the number of $-$ edges between clusters). Note that a .5 approximation for the maximization problem is trivial: either the partition of singletons or the single-cluster partition achieves this. Bansal et al. [2002] give a max-cut style PTAS for the complete, unweighted maximization problem, MaxAgree-Unweighted, and give an NP-completeness proof of the exact problem. Following work showed a .766-approximation to MaxAgree, the maximization problem on weighted, general (not necessarily complete) graphs [Swamy, 2004], both when the output partition can be arbitrary or must have size at most k. Their algorithm combines two approaches: first, a max-cut style SDP with a 2-hyperplane, Goemans-Williamson rounding, and second, they use the same SDP with a Frieze-Jerrum rounding using 6 hyperplanes. MaxAgree is APX-hard on general graphs [Charikar et al., 2003]. When there is a constant ratio between the maximum and minimum weight, the maximization problem admits a PTAS [Bonizzoni et al., 2008]. When the output is restricted to have at most k sets, each of MaxAgree-Unweighted-k and MinDisagree-Unweighted-k admits a PTAS [Giotis and Guruswami, 2006], in the complete graph setting. Recent work constructed a PTAS for each of MaxAgree-k and MinDisagree-k [Coleman and Wirth, 2010] when the weights are metric. Independently, Karpinski and Schudy [2009] construct a PTAS for MinDisagree-k (and a heirarchical version) which runs in time $n^2 2^{O(k^6/\epsilon^2)}$. The objective function is somewhat different in this work than in ours: again, the goal is to maximize the number of pairs (x, y) co-clustered if $w(x, y) = +1$ and anti-clustered if $w(x, y) = -1$.

Bansal et al. [2002] gave a constant-factor approximation for the minimization problem on complete, unweighted graphs, MinDisagree-Unweighted, and show a evidence for APX-hardness of MinDisagree. Charikar et al. [2003] improve this constant to 4 for MinDisagree-Unweighted; this constant was further improved to 3 by Ailon et al. [2008]. Three independent papers give $O(\log(|V|))$-approximations for MinDisagree [Charikar et al., 2003, Demaine and Immorlica, 2003, Emanuel and Fiat, 2003] and show its APX-hardness. Demaine et al. [2006] then shows that MinDisagree is APX-hard to approximate to a factor better than $\Omega(\log(|V|))$.

Consensus clustering There has been significant work on consensus clustering, which is defined as follows. The input to the problem is a collection of l partitions of the ground set $\Pi = \{\pi_1, \ldots, \pi_l\}$, where each $\pi_j : [n]^2 \to \{0, 1\}$ answers pairwise queries as to whether two elements are co-clustered. The goal of MinCon is to output a consensus cluster π such that $\sum_{j=1}^{l} d(\pi, \pi_j)$ is minimized (where $d(\pi, \pi')$ is the number of pairs $(a, b) \in [n]^2$ which are coclustered in one clustering but not in the other). An analagous maximization problem, MaxCon, can be defined in terms of maximizing the number of pairs on which the output clusterings agree (these problems coincide in the exact case; of course, they differ in their approximation algorithms and hardness). The general case of consensus clustering is known to be NP-hard [Kivnek and Morvek, 1986].

[Bonizzoni et al., 2008] give a PTAS for MaxCon, when there is no restriction on the size (or balance) of the output partition (using a polynomial integer program).

Considerably more work has been done on the minimization version of consensus clustering, MinCon. MinCon is $\frac{11}{7}$-approximable [Ailon et al., 2008] but APX-hard for $m > 2$ input partitions [Bonizzoni et al., 2008]. MinCon-k, when the size of the output partition is required to be at most $k \geq 2$, is NP-hard but admits a PTAS [Bonizzoni et al., 2009], based on a sampling of the elements $[n]$, inspired by the PTAS of Giotis and Guruswami [2006] for MinDisagree-k. Simultaneously to this work, Coleman and Wirth [2010] constructed a PTAS for MaxCon-k and MinCon-k (as a result of their PTAS for MaxAgree-k and MinDisagree-k with metric weights). MinCon remains NP-hard even when each input partition has at most 2 clusters when the output can have at most 2 clusters [Dörnfelder et al., 2011]. This implies, in particular, that the balanced consensus cut problem studied in Section 4 is NP-hard.

While the problems of MaxAgree and MaxCon (respectively, MinDisagree and MinCon) are quite similar, they are not directly comparable: the correlation problems are, in general, weighted, and certain instances of (weighted or unweighted) input graphs will not correspond to any collection of clusterings, since the triangle inequality need not hold for the correlation problems.

Maximum Correlation A third objective function has been studied in the literature, called MaxCorr [Giotis and Guruswami, 2006]. The objective value of an output clustering is the difference between the number of agreements and disagreements. On complete, weighted graphs, this has been shown to be approximable within a $O(\log(n))$ factor [Charikar and Wirth, 2004], and inapproximable within to a factor better than $\log^{\alpha}(n)$ for some $\alpha > 0$ [Arora et al., 2005].

Consensus Phylogenies The task of finding a consensus phylogeny is ubiquitous in biolinformatics and many popular phylogentic packages such as PHYLIP, Consense, Dendroscope, RadCon, and MESQUITE have built in heuristic routines for finding consensus subtrees among a given set of trees. An early survey of the area of constructing consensus phylogenies appears in Bryant [2003]. Closely related to our methods involving weighted matchings is the work on fast methods for comparing such trees Farach and Thorup [1994] and finding agreement trees Keselman and Amir [1994]. A unified treatment of such matching-based methods is in Kao et al. [2001]. Complexity results on comparing more than two trees are provided in Choy et al. [2005]. Related to our method of finding a single agreement supertree to match a given set of input trees that are not oriented at their root nodes, Jansson et al. [2005], Hoang and Sung [2011] present a method for finding rooted supertrees that have the highest compatibility with an input set of trees, defined using a very similar agreement function that we use in our work.

3 SDP Rounding for MaxBCorr

In this section, we present the proof of Theorem 2.

3.1 Integer program

We can describe the MaxBCorr problem using the following integer program. Here the Y variables indicate which split (and which side of that split) will be chosen from each family L_i to participate in the correlation objective, and the ± 1-valued X_j variables give the desired cut to be produced.

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{m} \sum_{(T,\bar{T})\in \mathrm{L}_i} \left(Y_{i,T} \sum_{j\in T} X_j + Y_{i,\bar{T}} \sum_{j\in \bar{T}} X_j\right) \\ \textstyle\sum_{(T,\bar{T})\in \mathrm{L}_i} Y_{i,T} + Y_{i,\bar{T}} &= 1 \quad \forall i \in [m] \\ \textstyle\sum_{j=1}^{n} X_j &= 0 \\ X_j &\in \{-1, 1\} \quad \forall j \in [n] \\ Y_{i,T}, Y_{i,\bar{T}} &\in \{0, 1\} \quad \forall i \in [m] \end{aligned} \tag{IP}$$

3.2 Semidefinite relaxation

We relax the above integer program to the following SDP:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{m} \sum_{T:(T,\bar{T})\in \mathrm{L}_i} Y_{i,T} \cdot Z_i \\ \|\textstyle\sum_{T\in \mathrm{L}_i} Y_{i,T} + Y_{i,\bar{T}}\| &= 1 \quad \forall i \in [m] \\ Y_{i,T} \cdot Y_{i,S} &= 0 \quad \forall i \in [m], \forall S \neq T \text{ s.t. } (S,\bar{S}), (T,\bar{T}) \in \mathrm{L}_i \\ \|\textstyle\sum_{j=1}^{n} X_j\| &= 0 \\ \|X_j\| &= 1 \quad \forall j \in [n] \\ Y_{i,T} \cdot \textstyle\sum_{j\in T} X_j \geq Y_{i,T} \cdot Z_i &\geq 0 \quad \forall i, \forall T : (T,\bar{T}) \in \mathrm{L}_i \end{aligned} \tag{SDP}$$

Note that there are vector variables $Y_{i,T}$ and $Y_{i,\bar{T}}$ for each split $(T,\bar{T})$ in each of the input families.

3.3 Rounding algorithm

Now, we describe the algorithm for rounding the SDP presented in Section 3.2. (This rounding will produce a cut that is only approximately-balanced; we will convert it into a perfectly balanced cut after the analysis.)

Algorithm 1: Rounding algorithm for the SDP

Data: The SDP solution vectors $\{X_j\}$
Result: An output split $\{-1,1\}^n$
1 Partition the X_j's using a random hyperplane;
2 Output the better of the two corresponding roundings (each halfspace goes to 1 or -1);

3.4 Analysis

For the output cut of the algorithm, we have to show that there are choices of splits from each family that give an approximately optimal agreement value.

To argue this, for the i'th family, we will pick a set T with probability $\|Y_{i,T}\|^2 = Y_{i,T} \cdot Y_{i,T}$ and consider the split $(T,\bar{T})$.

Let $\hat{X}_j \in \{-1,1\}$ and $\hat{Y}_{i,T} \in \{0,1\}$ indicate the rounded solution, i.e., the choice of cut and the choice of subset from each family of splits. Note that for any i, only one $\hat{Y}_{i,T}$ is 1 and the rest are 0.

Lemma 1. *Let x be a random variable in $[0,1]$ with $\mathsf{E}(x^2) = \alpha$. Then $\mathsf{E}(x) \geq \alpha$.*

Lemma 2.

$$\min_{\theta\in[0,\pi]} \frac{1 - \frac{2\theta}{\pi}}{\cos\theta} = \frac{2}{\pi}.$$

Lemma 3. *For each $i \in [m]$, and any T s.t. $Y_{i,T} \neq 0$, the SDP solution satisfies*

$$\|\sum_{j\in T} X_j\| \geq \|Z_i\|.$$

Proof. Fix $i \in [m]$. For each S s.t. $(S, \bar{S}) \in \mathrm{L}_i$, the SDP solution satisfies

$$Y_{i,S} \cdot \sum_{j \in T} X_j \geq Y_{i,S} \cdot Z_i \geq 0.$$

Let $Y^0_{i,S}$ be the unit vector in the direction of $Y_{i,S}$. Then it follows that

$$Y^0_{i,S} \cdot \sum_{j \in T} X_j \geq Y^0_{i,S} \cdot Z_i.$$

Let $\bar{X}_j$ be the projections of X_j to the span of $Y_{i,S}$. Then viewing $W = \sum_{j \in T} \bar{X}$ and Z_i in the orthonormal basis given by $Y^0_{i,S}$, the above inequality says that all coordinates of W and Z are nonnegative and each coordinate of W is at least as large as the corresponding coordinate of Z_i in this basis. It follows that $\|W\| \geq \|Z_i\|$. Moreover, since W is a projection of $\sum_{j \in T} X_j$, the lemma follows.□

We are now ready for the main lemma.

Lemma 4. *For any $i \in [m]$,*

$$\mathsf{E}\left(\left(\sum_{T:(T,\bar{T}) \in \mathrm{L}_i} \hat{Y}_{i,T} \sum_{j \in T} \hat{X}_j\right)^2\right) \geq \frac{2}{\pi}\left(\sum_{T:(T,\bar{T}) \in \mathrm{L}_i} Y_{i,T} \cdot Z_i\right)^2.$$

Proof.

$$\begin{aligned}
\mathsf{E}\left(\left(\sum_{T:(T,\bar{T}) \in \mathrm{L}_i} \hat{Y}_{i,T} \sum_{j \in T} \hat{X}_j\right)^2\right) &= \mathsf{E}\left(\sum_{T:(T,\bar{T}) \in \mathrm{L}_i} \hat{Y}^2_{i,T} \sum_{j,j' \in T} \hat{X}_j \hat{X}_{j'}\right) \\
&\text{(using the fact that } Y_{i,T} \text{ are orthogonal)} \\
&= \sum_{T:(T,\bar{T}) \in \mathrm{L}_i} \|Y_{i,T}\|^2 \sum_{j,j' \in T} \mathsf{E}\left(\hat{X}_j \hat{X}_{j'}\right) \\
&= \sum_{T:(T,\bar{T}) \in \mathrm{L}_i} \|Y_{i,T}\|^2 \sum_{j,j' \in T} \left(1 - \frac{\theta_{jj'}}{\pi}\right) - \frac{\theta_{jj'}}{\pi} \\
&= \sum_{T:(T,\bar{T}) \in \mathrm{L}_i} \|Y_{i,T}\|^2 \sum_{j,j' \in T} \left(1 - 2\frac{\theta_{jj'}}{\pi}\right) \\
&\geq \sum_{T:(T,\bar{T}) \in \mathrm{L}_i} \|Y_{i,T}\|^2 \sum_{j,j' \in T} \frac{2}{\pi} \cos\theta_{jj'} \\
&\text{(using Lemma 2)} \\
&= \frac{2}{\pi} \sum_{T:(T,\bar{T}) \in \mathrm{L}_i} \|Y_{i,T}\|^2 \sum_{j,j' \in T} X_j \cdot X_{j'} \\
&= \frac{2}{\pi} \sum_{T:(T,\bar{T}) \in \mathrm{L}_i} \|Y_{i,T}\|^2 \|\sum_{j \in T} X_j\|^2 \\
&\geq \frac{2}{\pi} \sum_{T:(T,\bar{T}) \in \mathrm{L}_i} \|Y_{i,T}\|^2 \|Z_i\|^2 \\
&\text{(using Lemma 3)} \\
&= \frac{2}{\pi} \| \sum_{T:(T,\bar{T}) \in \mathrm{L}_i} Y_{i,T}\|^2 \|Z_i\|^2 \\
&\geq \frac{2}{\pi} \| \sum_{T:(T,\bar{T}) \in \mathrm{L}_i} Y_{i,T} \cdot Z_i\|^2.
\end{aligned}$$

completing the proof.□

Lemma 5.

$$\mathsf{E}(f(\hat{X})) \geq \frac{2}{\pi} SDP \geq \frac{2}{\pi} OPT.$$

Proof. From Lemma 4, for each family i, the square of the agreement of the solution found by the rounding algorithm is at least $2/\pi$ times the square of the objective value for that family. Moreover, this bound is achieved by choosing the set T according to the $\|Y_{i,T}\|^2$. We can assume that the agreement is positive by choosing T or $\bar{T}$, whichever is higher. Next observe that

$$\sum_{T} \hat{X}_j + \sum_{\bar{T}} \hat{X}_j = \sum_{j=1}^{n} \hat{X}_j \geq 0.$$

(Since we try both $\hat{X}$ and $-\hat{X}$ in the rounding, the final quality is at least that of the choice where the sum is nonnegative). Therefore, using Lemma 1

$$\mathsf{E}(\max\{\sum_{T} \hat{X}_j, \sum_{\bar{T}} \hat{X}_j\} \geq \frac{2}{\pi} \sum_{T:(T,\bar{T})\in L_i} Y_{i,T} \cdot Z_i.$$

Adding up over all the m families proves the claim.□

We defer the proof of the following lemma to Appendix C.

Lemma 6. *Suppose* $\mathsf{E}[f(\hat{X})] \geq \frac{2}{\pi} OPT$. *Then, the greedy transformation of the output cut to a bisection [Frieze and Jerrum, 1995] will result in a bisection with expected correlation at least .611 times that of the optimal.*

Finally, we note that Theorem 2 is a direct corrollary of Lemmas 5 and 6. This entire analysis goes through for k-balanced cuts, $k = cn \leq \frac{n}{2}$. Furthermore, if one is sufficiently happy with an approximately balanced cut, our result gives a $\frac{2}{\pi} \approx .6366$-relative approximation, with the smaller side of the cut guaranteed to have at least $.878k$ vertices.

4 QPTAS for MaxBC and MinBC

In this section, we present a proof of Theorem 1. The algorithm we present for the maximization and minimization QPTASs are identical: in both cases, we will sample a small number of input families. For each family, try each set within it as the "representative": the one which will be compared with the output balanced cut to give the objective value in an optimal solution. Then, sort the elements $[n]$ according to their cost (benefit) when assigned label 1, and choose $n/2$ of them to label 1 in increasing (decreasing) order in terms of this cost (benefit).

The following two theorems state that Algorithm 2 is a QPTAS simultaneously for MinBC and MaxBC. Theorem 3 implies Theorem 1, our main minimization result.

Theorem 3. *Algorithm 2 computes a bisection S such that $\frac{c(S,\mathcal{F})}{c(S^*,\mathcal{F})} \leq (1+\epsilon)$, where S^* is the minimum cost consensus cut for input $\mathcal{F}$, in time $(2N)^{O(\log(n)/\varepsilon^2)}$, if $|F_j| \leq N$ for all $j \in [m]$.*

Theorem 4. *Algorithm 2 computes a bisection S such that $\frac{A(S,\mathcal{F})}{A(S^*,\mathcal{F})} \geq (1-\epsilon)$, where S^* is the cut which maximizes agreement for input $\mathcal{F}$, in time $(2N)^{O(\log(n)/\varepsilon^2)}$, if $|F_j| \leq N$ for all $j \in [m]$.*

Before we prove Theorems 3 and 4, we present a lemma which will be useful in proving the maximization guarantee as well. Recall that the cost of a label on an element is a number between zero and one denoting the fraction of the input families with which we have disagreement (agreement in the maximization version).

Algorithm 2: $(1+\epsilon)$-approximation to MinBC, or $(1-\epsilon)$-approximation to MaxBC

Data: m families $\mathcal{F} = \{(F_1^1, \bar{F}_1^{k_1}), \ldots, (F_m^1, \bar{F}_m^{k_m})\}$, $\cup_k F_j^k = [n]$ for all j
Result: Cut $S \subset [n]$ such that $|S| = n/2$

Let T be a set of families of size $s = O(\log n/\varepsilon^2)$ chosen u.a.r.;
Relabel families such that $F_1, \ldots, F_T$ is the sample;
for *each choice of* $(t_1, \ldots, t_T)$*, where* $t_j \in [k_j]$ *for each* $j \in [T]$ **do**
    (*Choose a set from each family*);
    for *each choice of* $x \in \{-1, 1\}^T$ **do**
        (*Choose* $F_j^{t_j}$ for $x_j = 1$ or $\bar{F}_j^{t_j}$ for $x_j = -1$*);
        (*Count # of aligned samples' sets which label* i *with a* 1*);
        Let $n(i) = \sum_{k=1}^{T} \mathbb{I}[(i \in F_k^{t_k} \wedge x_k = 1) \vee (i \in \bar{F}_k^{t_k} \wedge x_k = -1)]$;
        $S_{t_1,\ldots,t_T,x}$ = the $n/2$ elements such that $n(i)$ is largest;
        Let $c(t_1, \ldots, t_T, x) = c(\mathcal{F}, S_{t_1,\ldots,t_T,x})$
Let $\overline{(t_1, \ldots, t_T, x)} = argmin_{t_1,\ldots,t_T,x} c(t_1, \ldots, t_T, x)$;
Output $S_{\overline{(t_1,\ldots,t_T,x)}}$;

Lemma 7. *Let* $\tilde{c}(i,x)$ *denote the* cost on the sample of label x for element i, when we have chosen the sets that OPT chooses for the sample*, and* $c(i,x)$ *is the cost on the entire input according to the sets which OPT chooses. Then, with probability at least* 9/10*, for all* i *and for all* x*,* $\tilde{c}(i,x) \in [c(i,x) - \epsilon', c(i,x) + \epsilon']$ *for some fixed* $\epsilon' > 0$*.*

Proof. We claim that we approximate, according to the sample and the optimal choice of sets for the sample, the actual optimal cost of assigning element i a label either 0 and 1 within $\epsilon' = O(\epsilon)$, for all i, with probability at least 9/10. This holds for a fixed i and label 0 or 1 by a simple Hoeffding bound, with probability at least $1 - \frac{1}{20n}$, on a sample of size $O\left(\log(n)/\epsilon^2\right)$. Thus, by a union bound, this will hold for all $2n$ element/label pairs, with probability at least 9/10.□

Theorem 4 for the maximization objective follows directly from an application of Lemma 7 together with the fact that the agreement of OPT is always at least $n/4$: an additive $\epsilon' = \epsilon/4$ per edge will only add cost $\epsilon' n$ additively, implying a $(1+\epsilon)$ multiplicative guarantee.

On the other hand, the minimization result is considerably more involved: multiplicative guarantees of small quantities require more care. However, when the optimal cost is quite small, many elements will have one label which looks inexpensive on the sample and is inexpensive on the sample as well. Without further ado, we proceed with the proof of Theorem 3.

Proof. Consider the time which we pick, for the sample T, the sets for each element of T corresponding to the sets used on T when compared to l^*, the optimal bisection. For the remainder of this argument, we will focus only on our estimation with respect to this choice.

Now, we make a more refined matching argument about l^* which yields a multiplicative guarantee rather than the additive one given simply by applying the above per-element additive guarantee given by Lemma 7. Consider the bipartite graph with n nodes on the left corresponding to the elements, and n nodes on the right (the first $n/2$ corresponding to the label "0" and the next $n/2$ corresponding to the label "1"). Let edge weight $w'(i,x)$ corresponds to the cost of labeling i with x, calculated on the sample (for the optimal choice of sets for each family in the sample). We will argue about the matchings on the graph with edge weights $w(i,x) = \max(0, w'(i,x) - \epsilon')$: this does not change the ordering of the elements by their cost (and thus does not change the output of the algorithm, which outputs a min-cost matching with edge weights w'). This allows us to be certain that our estimated costs are *underestimates*.

Let $\tilde{c}$ denote the cost of a matching with respect to the graph with edge-weights w and c denote the cost of a matching with respect to the *actual cost of the matching according to the entire input aligned according to OPT*. Let l' be the matching that our algorithm selects. It is clear that $\tilde{c}(l') \leq \tilde{c}(l^*)$; our algorithm chose a min-cost matching with respect to its percieved costs.

Now define the set $M = \{(i,x)|\tilde{c}(i,x) \leq 1/4 - \epsilon'\}$, the set of pairs of elements and labels for which that labeling of that element has estimated cost less than $1/4 - \epsilon'$. Notice that if $(i,x) \in M$, then $\tilde{c}(i,x') \geq 3/4$: a given element has at most one label which is quite inexpensive, and the other will look quite expensive. Moreover, each pair $(i,x) \notin M$ implies that $c(i,x) \geq 1/4$, so l^* can use at most $4c(l^*)$ of these. Similarly, l' can use at most $4c(l^*)$ of these "expensive" edges, or we arrive at a contradiction to the optimality of l' with respect to $\tilde{c}$:

$$\tilde{c}(l') \geq c(l^*) \geq \tilde{c}(l^*)$$

where the first inequality comes from the fact that having more than $4X$ edges of weight at least $1/4$ implies cost at least X, and the second comes from the fact that all the estimated costs are lower bounds on the actual costs. Thus, since there are at most two *labels* for a given element, and at most one of them is inexpensive, for all but $16c(l^*)$ elements, l and l' use the same labels, and *pay exactly the same amount!* Thus, we have

$$\begin{aligned} c(l') - c(l^*) &\leq \epsilon' |l' \setminus l^*| \\ &\quad \text{(If not, } \tilde{c}(l^*) < \tilde{c}(l') \text{, by Lemma 7 a contradiction)} \\ &\leq 16\epsilon' c(l^*) \\ &\quad \text{(Since } l' \text{ uses at most } 16c(l^*) \text{ edges } l^* \text{ does not)} \\ &\leq \epsilon c(l^*) \end{aligned}$$

Thus, $\frac{c(l')}{c(l^*)} \leq (1+\epsilon)$, and this completes the proof.□

5 Exact algorithm when the input is a collection of fully aligned trees

We now present a version of the problem of reconciling phylogenetic trees for which we can achieve exact optimality in polynomial time.

Specifically, suppose each input tree has a root, and in addition each split is labeled as L or R according to whether it corresponds to a left or right subtree of its parent respectively. In this setting, the input trees can be thought of as mappings from the elements $1, \ldots, n$ to L/R-strings. In this case, a natural notion of the *cost* of a given output labeling l (assigning each element $i \in [n]$ to a string $l(i) \in \{L,R\}^n$), is:

$$c_{\text{LCA}}(l, \mathcal{T}) = \sum_{T_j \in \mathcal{T}} \sum_{i=1}^{n} d_{\text{LCA}}(l(x), T_j(x))$$

where $d_{\text{LCA}}(x,y)$ is the distance between x, y according to their least common ancestor according to a lexicographic path: for example, $d_{\text{LCA}}(LRLL, LR) = 2$, $d_{\text{LCA}}(LRRRL, LRRL) = 3$. We call this problem the **Minimum-Cost Aligned Tree** problem. In this section, we present a polynomial-time algorithm which computes the labeling l^* with minimum cost with respect to c_{LCA}.

The algorithm above can be thought of as follows. It takes the topologies of all the input trees, unions them together, adds all nodes on the shortest path between elements (all internal nodes in the union), and then adds additional nodes to ensure each node in this topology has a $\log(n)$-depth complete binary tree beneath it. We call this tree the scaffolding tree. It then solves for a min-cost matching between the elements and the nodes in this scaffolding tree.

Theorem 5. *Algorithm 3 computes a min-cost labeling for the input trees $T_1, \ldots, T_m$, with the objective function $c(l) = \sum_T \sum_{i=0}^{n} d_{\text{LCA}}(l(x), T(x))$.*

Algorithm 3: Minimum-cost aligned tree: min-cost consensus tree when input is a collection of aligned trees (mapping elements to L/R strings)

Data: m trees $T_1, \ldots, T_m : [n] \to \{L, R\}^n$
Result: Labeling $l : [n] \to \{L, R\}^n$
Let $\hat{S} = \cup_{j \in [m]} Range(T_j) \cup \{y | x \in T_j, x' \in T_{j'}, y \in Path(x, x')\}$;
Let $S = \{x \cdot y | x \in \hat{S}, y \in \{L, R\}^t, 0 \leq t \leq \log(n)\}$;
For each $i \in [n]$ and each $r \in S$, let $c(i, r) = \sum_{T_j} d_{\text{LCA}}(r, T(i))$ be the cost of labeling i with label r;
$l =$ the min-cost matching of $[n]$ to L according to c;

Proof. Assume, without loss of generality, that there is at least one input tree that begins some label with L and at least one input tree that begins some label with R. If this is not the case, without loss of generality we can assume all labels begin with the same letter and can think of the reduced problem on the subtree.

We claim the topology S contains an optimal topology. Notice that, if this is the case, our algorithm is correct; we compute the min-cost labeling within this topology.

Now, we prove our claim. Suppose for contradiction that no optimal tree is contained entirely within the topology S. Consider one particular optimal labeling l^*. There is some $i \in [n]$ such that $l^*(i) \notin S$ by assumption. Notice that $l^*(i)$ must be a decendant of some $r \in \hat{S}$: in particular, l^* is a descendant of the all-empty string (which, by our assumption that some tree has a nontrivial top split, is in $\hat{S}$).

Consider the ancestor $r \in \hat{S}$ of $l^*(i)$ which is closest to $l^*(i)$. It must be the case that $||l^*(i) - r||_1 > \log(n)$, or $l^*(i)$ would be in the S (since we've added a complete binary tree of that depth beneath r). But then, it would be strictly cheaper to move $l^*(i)$ into this complete binary tree beneath r: for each input tree T_j, $T_j(i)$ is closer to r than any descendant of r. Thus, decreasing the distance between r and the descendant $l^*(i)$ will decrease the total cost for i for each input tree. Moreover, there is some unoccupied space beneath r (there are n nodes beneath r in this tree with distance at most $\log(n)$ from r).□

In Appendix A, we show how to extend this result to get a $QPTAS$ for the case where the input trees' cuts are labeled with L/R everywhere but the topmost level.

References

Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)*, 55(5):23, 2008.

Sanjeev Arora, Eli Berger, Elad Hazan, Guy Kindler, and Muli Safra. On non-approximability for quadratic programs. In *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, pages 206–215. IEEE, 2005.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. In *Foundations of Computer Science, 2002. Proceedings. 43th Annual IEEE Symposium on*, pages 238–247, 2002.

Paola Bonizzoni, Gianluca Della Vedova, Riccardo Dondi, and Tao Jiang. On the approximation of correlation clustering and consensus clustering. *Journal of Computer and System Sciences*, 74(5):671–696, 2008.

Paola Bonizzoni, Gianluca Della, and Vedova Riccardo Dondi. A ptas for the minimum consensus clustering problem with a fixed number of clusters. In *In Proc. 11th ICTCS*. Citeseer, 2009.

David Bryant. A classification of consensus methods for phylogenetics. 2003.

Moses Charikar and Anthony Wirth. Maximizing quadratic programs: extending grothendieck's inequality. In *Foundations of Computer Science, 2004. Proceedings.*, pages 54–60. IEEE, 2004.

Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 524–533. IEEE, 2003.

Charles Choy, Jesper Jansson, Kunihiko Sadakane, and Wing-Kin Sung. Computing the maximum agreement of phylogenetic networks. *Theoretical Computer Science*, 335(1): 93 – 107, 2005. ISSN 0304-3975. doi: http://dx.doi.org/10.1016/j.tcs.2004.12.012. URL

http://www.sciencedirect.com/science/article/pii/S0304397504008102. Pattern Discovery in the Post Genome.

Tom Coleman and Anthony Wirth. A polynomial time approximation scheme for k-consensus clustering. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 729–740. Society for Industrial and Applied Mathematics, 2010.

Erik D Demaine and Nicole Immorlica. Correlation clustering with partial information. In *Approximation, Randomization, and Combinatorial Optimization.. Algorithms and Techniques*, pages 1–13. Springer, 2003.

Erik D Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361(2):172–187, 2006.

Martin Dörnfelder, Jiong Guo, Christian Komusiewicz, and Mathias Weller. On the parameterized complexity of consensus clustering. In *Algorithms and Computation*, pages 624–633. Springer, 2011.

Dotan Emanuel and Amos Fiat. Correlation clustering–minimizing disagreements on arbitrary weighted graphs. In *Algorithms-ESA 2003*, pages 208–220. Springer, 2003.

Martin Farach and Mikkel Thorup. Fast comparison of evolutionary trees. In *Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 481–488. Society for Industrial and Applied Mathematics, 1994.

Alan Frieze and Mark Jerrum. Improved approximation algorithms for max k-cut and max bisection. In *Integer Programming and Combinatorial Optimization*, pages 1–13. Springer, 1995.

Ioannis Giotis and Venkatesan Guruswami. Correlation clustering with a fixed number of clusters. In *Proceedings of the seventeenth annual ACM-SIAM SODA*, pages 1167–1176. ACM, 2006.

VietTung Hoang and Wing-Kin Sung. Improved algorithms for maximum agreement andcompatible supertrees. *Algorithmica*, 59(2):195–214, 2011. ISSN 0178-4617. doi: 10.1007/s00453-009-9303-6. URL http://dx.doi.org/10.1007/s00453-009-9303-6.

Jesper Jansson, Joseph H.-K. Ng, Kunihiko Sadakane, and Wing-Kin Sung. Rooted maximum agreement supertrees. *Algorithmica*, 43(4):293–307, 2005. ISSN 0178-4617. doi: 10.1007/s00453-004-1147-5. URL http://dx.doi.org/10.1007/s00453-004-1147-5.

Ming-Yang Kao, Tak-Wah Lam, Wing-Kin Sung, and Hing-Fung Ting. An even faster and more unifying algorithm for comparing trees via unbalanced bipartite matchings. *Journal of Algorithms*, 40(2):212–233, 2001.

Marek Karpinski and Warren Schudy. Linear time approximation schemes for the gale-berlekamp game and related minimization problems. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 313–322. ACM, 2009.

Dmitry Keselman and Amihood Amir. Maximum agreement subtree in a set of evolutionary trees-metrics and efficient algorithms. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 758–769. IEEE, 1994.

Mirko Kivnek and Jaroslav Morvek. Np-hard problems in hierarchical-tree clustering. *Acta Informatica*, 23 (3):311–323, 1986. ISSN 0001-5903. doi: 10.1007/BF00289116.

Srinath Sridhar, Kedar Dhamdhere, Guy E. Blelloch, Eran Halperin, R. Ravi, and Russell Schwartz. Algorithms for efficient near-perfect phylogenetic tree reconstruction in theory and practice. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 4(4):561–571, 2007a.

Srinath Sridhar, Fumei Lam, Guy E. Blelloch, R. Ravi, and Russell Schwartz. Direct maximum parsimony phylogeny reconstruction from genotype data. *BMC Bioinformatics*, 8, 2007b.

Chaitanya Swamy. Correlation clustering: maximizing agreements via semidefinite programming. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 526–527. Society for Industrial and Applied Mathematics, 2004.

Ming-Chi Tsai, Guy E. Blelloch, R. Ravi, and Russell Schwartz. A consensus tree approach for reconstructing human evolutionary history and detecting population substructure. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 8(4):918–928, 2011.

Richard T Wong. Worst-case analysis of network design problem heuristics. *SIAM Journal on Algebraic Discrete Methods*, 1(1):51–63, 1980.

A Minimization Problems

QPTAS when the input is a collection of almost-aligned trees and an output topology Assume that each input tree is labeled L/R at each cut except for the one from the root; that is, for each j, $T_j : [n] \to \{?_x, ?_y\} \times \{L, R\}^*$.The $?_x, ?_y$ characters can be disambiguated (e.g., $?_x \neq ?_y$ in the same tree), but aren't prespecified to be L or R: for example, if the input is two otherwise identical trees T_1, T_2 with T_1's labels for a set X beginning with $?_x$ and $?_y$ for a set Y, and the other tree's labels of X begin with $?_y$ (and Y with $?_x$), then either tree is distance 0 from T_1 where $?_x = \{L, R\}$ and $?_y = \neg ?_x$ are substituted.

Furthermore, assume the input includes a topology with n location into which the elements are supposed to be placed. The objective function for an output labeling l is as follows:

$$c(l) = \sum_{T} \min_{T' = T_L, T_R} \sum_{i=0}^{n} d_{\text{LCA}}(l(x), T'(x))$$

l here is a matching of elements to locations in the output topology O (equivalently, locations can be thought of as a sequence of Ls and Rs). If the $l(x)$ and $T'(x)$ are of different length, the prefixes will be compared for their distance, and the difference in length will be added to that distance.

We present an algorithm which gives an ϵnm-additive approximation guarantee to the optimal output labeling, in time $n^{O(d^2/\varepsilon^2)}$.

Our algorithm is as follows.

Algorithm 4: Algorithm for computing an approximately optimal consensus, input are almost-aligned trees and a fixed output topology

Data: $T^1, \ldots, T^m$, almost-aligned trees, and an output topology O

Pick $s = \frac{d^2}{2\epsilon^2}(\ln(20) + 2\ln(n)))$ random input trees T^i;

for *each possible set of alignments* $A \in \mathcal{A}$*, i.e.,* T^i_L *or* T^i_R *for each* i **do**

  Let m_A be the min-cost matching on O w.r.t. the sample;

  Let $c(m_A)$ = the cost of m_A w.r.t the full data set; *once a matching is fixed, aligning each T can be done separately

Output the m_A with the minimum cost among all 2^s candidates;

Theorem 6. *The above algorithm runs in time $n^{O(d^2/\varepsilon^2)}$. Furthermore, it gives an ϵnm additive approximation algorithm for output topologies of depth at most d.*

Proof. For the purposes of this argument, when we talk about cost of a matching, we mean the *average* cost of this matching per input tree.

Consider a location t in the output topology and an element x. We claim that, for all t and x, the cost of placing x at t will be well-approximated according to the sample S and the alignment according to OPT; then, this will imply the min-cost matching with respect to the sample should only have slightly larger cost with respect to all the input trees, and also that the optimal labeling l^* with respect to all input trees should won't have cost too much smaller than what it appeared to have on the sampled trees. Together, these imply that the output labeling won't have cost too much larger than l^*.

For a given t, x, let $C_{x,t}$ be the average (over input trees) cost of x at location t according to the optimal alignment. Now, for a particular input tree T^i, let C_{x,t,T^i} denote the cost of x at location t w.r.t the input T^i aligned as according to OPT. Then, Hoeffding's inequality will imply that $\mathsf{P}[|\frac{1}{s}\sum_{i \in S} C_{x,t,T^i} - C_{x,t}| \geq \epsilon] \leq 2e^{-\frac{2\epsilon^2 s}{d^2}}$. For $s \geq \frac{d^2}{2\epsilon^2}(\ln(20) + 2\ln(n)))$ samples, the event will happen with probability at most $\frac{1}{10n^2}$ for a particular location/element pair. Using a union bound, with probability 9/10, no element/location pair's optimally-aligned cost on the sample will be more than an additive ϵ from its optimally-aligned cost on the entire input.

In particular, this implies that l^*, the min-cost matching with respect to OPT and the entire input will cost at most ϵn more on the sample, and also that l', the min-cost matching with respect to the optimal alignment on the sample, will have cost at most ϵn more on the entire input. Let $\tilde{c}$ be the cost of a matching with respect to the optimal alignment on the entire input, and $\hat{c}$ be the cost of a matching w.r.t. the optimal alignment on the sample. Since l' was the min-cost matching with respect to the optimal alignment on the sample, the above probabalistic argument tells us that

$$\tilde{c}(l^*) + \epsilon n \geq \hat{c}(l^*) \geq \hat{c}(l') \geq \tilde{c}(l') - \epsilon n$$

which, rearranged, implies that $\tilde{c}(l') \leq \tilde{c}(l^*) + 2\epsilon n$. Now, consider the matching $\hat{l}$ that our algorithm output. Let c be the cost of a matching with respect to *its* optimal alignment on the entire input. Then, we know that

$$c(l^*) = \tilde{c}(l^*) \geq \tilde{c}(l') - 2\epsilon n \geq c(l') - 2\epsilon n \geq c(\hat{l}) - 2\epsilon n$$

The first equality comes from the fact that l^* was optimal with respect to the optimal alignment, the first inequality comes from above, the second inequality comes from the fact that the optimal alignment for l' only costs less than the optimal alignment for l^*, and the final inequality comes from the fact that $\hat{l}$ was chosen because it had lower cost on the entire input than did l' (w.r.t their optimal alignments). Summing up over all m input trees gives the desired result.

In fact, a very minor modification to the above algorithm gives an $(1+\epsilon)$-multiplicative guarantee. Suppose, prior to computing the min-cost matching for a fixed alignment on the sample, each edge-weight $w(i,t)$ computed with respect to the alignment A was replaced with $w'(i,t) = \max\{0, w(i,t) - \epsilon\}$. This ensures that the algorithm is only *underestimating* the cost of the matching it outputs: in particular, it will be sufficient to imply that the estimated cost of the output matching is less than the true cost of the optimal matching.

Theorem 7. *Let $\epsilon' = \frac{\epsilon}{16}$. Algorithm 4, running with ϵ', with costs shifted by ϵ', gives a $(1+\epsilon)$-approximates OPT. It runs in time $n^{O(d^2/\varepsilon^2)}$ for input trees $T_1, \ldots, T_m$ of depth at most d, for a fixed output topology.*

Before we prove the main theorem, we make a small observation which is crucial to the proof. Simply put, once a sample and an alignment is fixed, if a given location is inexpensive for a particular element, all other locations must have noticeably higher cost for that element, and all other elements must have noticeably higher cost for that location.

Lemma 8. *Fix an alignment A and a sample S. Then, if $c(t,i) \leq x < \frac{n}{2} - \epsilon'$, for all $i' \neq i$, $c(t,i') \geq n - x > \frac{n}{2} + \epsilon'$, and for all $t' \neq t$, $c(t',i) \geq n - x > \frac{n}{2} + \epsilon'$.*

Proof. If the cost of (i,t) is $x < \frac{n}{2} - \epsilon'$, this implies that more than half of the input trees place i at exactly t. This implies any other location t' for i will have cost at least 1 for at least $n - x$ input trees, and similarly for any other i' at t.

Now, we proceed with the proof of Theorem 7.

Proof. Consider the optimal alignment A^* and the optimal matching l^*. Now, on the sample drawn, consider the alignment that corresponds to A^* on the sample. If l^* is chosen, the algorithm 1−approximates OPT. On the other hand, if the algorithm choose l' for the min-cost matching on the sample aligned with A^*, we need to argue that $\frac{c(l')}{c(l^*)} \leq (1+\epsilon)$, where c is the cost on the entire input, averaged over the number of input trees.

Now, consider the bipartite graph with n nodes on the left corresponding to the elements, and n nodes on the right corresponding to locations, for alignment A^*. If the edge weight $w(i,t)$ corresponds to the cost of this edge w.r.t the sample aligned as A^*, let $w'(i,t) = \max\{0, w(i,t) - \epsilon'\}$. For the remainder of this argument, we will argue about the graph with edge weights w'. Now, let $\tilde{c}$ denote the cost of a matching with respect to the graph with edge-weights w'. Let l' be the matching that our algorithm selects.

From this, it is clear that $\tilde{c}(l') \leq \tilde{c}(l^*)$; our algorithm chose a min-cost matching. Moreover, since we've only underestimated costs, $\tilde{c}(l^*) \leq c(l^*)$. Thus, $\tilde{c}(l') \leq c(l^*)$ (the algorithm's shifted perceived cost on the sample of l' is less than the true cost of l^*).

Now, by Lemma 8, the edges with cost at most 1/4 w.r.t w' form a (not necessarily perfect) matching μ. It is clear that OPT uses at most $4\tilde{c}(l^*)$ edges which are not part of μ. Similarly, since $\tilde{c}(l') \leq \tilde{c}(l^*)$, l' uses at most $4\tilde{c}(l^*)$ edges which are not part of μ. Thus, l' and l^* can disagree on at most $8\tilde{c}(l^*)$ edges in μ and $8\tilde{c}(l^*)$ edges outside of μ. Thus, the total number of edges l' uses but l^* does not use is at most $16\tilde{c}(l^*)$. Then, if we look at the *true* cost of these labelings with respect to A^* and the entire input, we have

$$
\begin{aligned}
c(l') - c(l^*) &\leq \epsilon' |l' \setminus l^*| \\
&\quad \text{(Theorem 6 guarantees } \epsilon'\text{-additive error per edge)} \\
&\leq \epsilon' |l' \setminus l^*| \\
&\leq 16\epsilon' \tilde{c}(l^*) \\
&\quad \text{(Since } l' \text{ uses at most } 16\tilde{c}(l^*) \text{ edges } l^* \text{ does not)} \\
&\leq \epsilon \tilde{c}(l^*) \\
&\leq \epsilon c(l^*) \\
&\quad \text{(Since } \tilde{c}(l^*) \leq c(l^*)\text{)}
\end{aligned}
$$

Thus, $\frac{c(l')}{c(l^*)} \leq (1+\epsilon)$, and this completes the proof.

The ideas in Section 5 in fact show us how to construct an output topology which is guaranteed to have an optimal topology within it: namely, the scaffolding topology for both alignments of each input tree, built out to complete trees as before.

Theorem 8. *Suppose each input tree has no degree-2 nodes, and is depth at most d. Then, the scaffolding tree for the* $2m$ *trees (each input aligned both left and right) contains an optimal topology. Thus, sampling at most* $O(\frac{d^2 \ln(n^3 m)}{\epsilon^2})$ *to guarantee* $1+\epsilon$ *multiplicative error.*

Proof. If no input tree has a degree-2 node, in particular this implies there are at most $2n$ nodes in either of its alignments. Thus, there are at most $4n$ locations in both of its alignments, and at most $4n^2$ once each node has been given a complete binary tree of depth $\log(n)$ beneath it. Thus, the scaffolding tree in total will have at most $4n^2 m$ locations for which one will need to estimate the cost of each of n elements; by an analagous union bound from Theorem 6, this sample size will be large enough to estimate the costs of each element to each location with additive precision ϵ. By an identical argument to the one in Theorem 5, this scaffolding tree contains an optimal topology. Thus, by Theorem 7, the additive per-edge error guarantee will suffice to construct a $(1+\epsilon)$-approximation to OPT.

B Chains to permutations: an exact solution

Suppose the input to our problem is a collection of chains: the laminar family representation of the input phylogenies is just a set of inclusive subsets. Formally, our input is

1. A set of m weighted laminar families $\mathrm{L}_1, \ldots, \mathrm{L}_m$, each family being a collection of weighted subsets of $\{1, \ldots, n\}$; e.g. $L_i = \{(w_1, S_1), (w_2, S_2), \ldots, (w_{k_i}, S_{k_i})\}$

   such that, for each $j \in [m]$, for each $(w, S), (w', S') \in L_j$, either $S \subseteq S'$ or $S' \subseteq S$.
   Our output is then a permutation σ of the n ground set elements.

The quality of a solution is defined as follows.

$$c(\sigma) = \sum_{i=1}^{m} \sum_{j=1}^{k_i} w_j(\|S_j - \sigma_{|S_j|}\|_1).$$

where σ_t is the t-prefix of the permutation. That is, the cost of σ with respect to a particular laminar family will charge the weighted L_1 distance from each set on the family to the prefix of equal size in σ.

The above problem reduces to a min-cost perfect matching problem. Namely, there are n items which need to be matched into n locations. For a fixed item and a fixed location, there is a cost (namely, the number of charges corresponding to the weighted L_1 distances that element takes part in). Formally, the cost incurred for item e in location l, $c_{e,l}$, is just

$$c_{e,l} = \sum_{i=1}^{m} \sum_{j=1}^{k_i} w_j \mathbb{I}[e \in S_j \text{ and } l > |S_j|]$$

Thus, we can find a solution of minimum cost in polynomial time.

C Balance for the SDP rounding

Here, we prove Lemma 6 by presenting a Frieze-Jerrum style analysis of the balance condition of the rounded solution.

Lemma 9. *Suppose $k = cn$ for some constant $1/2 \geq c \geq 0$. The random hyperplane gives a cut which has at least an $\alpha_2 c$-fraction of the nodes on the smaller side of the cut (when c is the fraction of nodes on the smaller side of an exact solution, and $\alpha_2 \approx .878$, the max-cut constant).*

Proof. Let S_t be a random variable which takes on the set of j such that $\hat{X}_j = 1$, and without loss of generality assume $|S_t| \geq \frac{n}{2}$. Following the analysis of Frieze-Jerrum [Frieze and Jerrum, 1995], we wish to analyze $\mathsf{E}[|S_t|(n - |S_t|)]$:

$$
\begin{aligned}
\mathsf{E}[|S_t|(n-|S_t|)] &= \sum_{i<j} \phi(X_i \cdot X_j) \\
&\geq \frac{\alpha_2}{2} \sum_{i<j} (1 - X_i \cdot X_j) \\
&\geq \frac{\alpha_2}{2} \left(\frac{n(n-1)}{2} - \sum_{i<j} X_i \cdot X_j \right) \\
&\geq \frac{\alpha_2}{2} \left(\frac{n(n-1)}{2} - \sum_{i<j} X_i \cdot X_j - \sum_i \frac{X_i \cdot X_i}{2} + \sum_i \frac{X_i \cdot X_i}{2} \right) \\
&= \frac{\alpha_2}{2} \left(\frac{n(n-1)}{2} - \frac{1}{2} || \sum_i X_i ||^2 + \sum_i \frac{X_i \cdot X_i}{2} \right) \\
&= \frac{\alpha_2}{2} \left(\frac{n(n-1)}{2} - \frac{1}{2} || \sum_i X_i ||^2 + \frac{n}{2} \right) \\
&\text{(Since } ||X_i|| = 1) \\
&= \alpha_2 \left(\frac{n^2}{4} - \frac{1}{4}(2k-n)^2 \right) \\
&\text{(By } || \sum_i X_i || = 2k - n = 2cn - n) \\
&= \alpha_2 \left(\frac{n^2}{4} - \frac{1}{4}(2cn-n)^2 \right) \\
&= \alpha_2 n^2 c(1-c)
\end{aligned}
$$

thus completing the proof.□

With Lemma 9 in hand, it is possible to convert the $\frac{2}{\pi}$-approximation algorithm which outputs a α_2-balanced cut to an approximation algorithm which gives an exact bisection. This is done in exactly the same way as in Frieze and Jerrum [1995], by removing vertices from larger set in increasing order w.r.t. the number of edges across the cut. That is, let $S_t = \{x_1, \ldots, x_l\}, l > n/2$, without loss of generality. Then, let $\eta(x_i) = w(x_i, \overline{S_t})$, and $\eta(x_1) \geq \eta(x_2) \geq \ldots \geq \eta(x_l)$. Then, output $\tilde{S}_t = \{x_1, \ldots, x_{n/2}\}$.

Theorem 9. *The greedy modification of the SDP rounding algorithm described above guarantees a .611137-approximation and outputs a bisection.*

Proof. Let $V_t = |S_t|(n - |S_t|)$; by Lemma 9, we have that

$$\mathsf{E}[V_t] \geq \frac{\alpha_2 n^2}{4}. \tag{1}$$

Let $G_t = w(S_t : V \setminus S_t)$, the weight of the rounded cut. Lemma 5 implies that

$$\mathsf{E}[G_t] \geq \frac{2}{\pi} OPT \tag{2}$$

If, as in Frieze and Jerrum [1995], we define

$$Z_t = \frac{V_t}{\frac{1}{4}n^2} + \frac{G_t}{OPT} \tag{3}$$

it follows that $Z_t \leq 2, E[Z_t] \geq \alpha_2 + \frac{2}{\pi}$. Thus, if we consider p, the probability that Z_t could be more than a $(1-\epsilon)$-factor from its expecation, we have

$$p(1-\epsilon)\mathsf{E}[Z_t] + (1-p)2 \geq \mathsf{E}[Z_t]$$

which implies $p \leq \frac{1-\mathsf{E}[Z_t]/2}{1-(1-\varepsilon)\mathsf{E}[Z_t]/2}$. Thus, after $K = poly(\frac{1}{\epsilon}$ repititions, we have that $p^K \leq \epsilon$, and so we may assume that $Z_t \geq (1-\epsilon)\mathsf{E}[Z_t]$ for the remainder of the analysis. This allows us to lower-bound V_t in terms of G_t; suppose $G_t = \lambda OPT$. We have

$$V_t \geq n^2/4\left((1-\epsilon)\mathsf{E}[Z_t] - \lambda\right)$$

from 3. If $|S_t| = \delta n$, we have

$$\delta(1-\delta) \geq \frac{1}{4}\left((1-\epsilon)\mathsf{E}[Z_t] - \lambda\right)$$

which implies

$$\lambda \geq (1-\epsilon)\mathsf{E}[Z_t] - 4\delta(1-\delta) \tag{4}$$

But then,

$$\begin{aligned}
& w(\tilde{S_t}; V \setminus \tilde{S_t}) && (5)\\
& \geq \frac{nw(S_t; V \setminus S_t)}{l} && (6)\\
& = \frac{w(S_t)}{2\delta} && (7)\\
& = \frac{\lambda OPT}{2\delta} && (8)\\
& \geq \frac{(1-\epsilon)\mathsf{E}[Z_t] - 4\delta(1-\delta)}{2\delta} OPT && (9)\\
& \geq \frac{\mathsf{E}[Z_t](1-\varepsilon) - 2\sqrt{2}\left(1 - \sqrt{\frac{\mathsf{E}[Z_t](1-\varepsilon)}{2}}\right)\sqrt{\mathsf{E}[Z_t](1-\epsilon)}}{\sqrt{2\mathsf{E}[Z_t](1-\varepsilon)}} OPT && (10)\\
& \approx .611137 OPT && (11)
\end{aligned}$$

where the second to last line follows from basic calculus.□