Report

Hunting for functionally analogous genes

Author(s):
Hallett, M.T.; Lagergren, Jens

Publication Date:
1999

Permanent Link:

https://doi.org/10.3929/ethz-a-006653414 →

Hunting for Functionally Analogous Genes

M. T. Hallett[1⋆] , J. Lagergren[2]

[1] Computational Biochemistry Research Group
Dept. of Computer Science, ETH Zürich, Zürich, Switzerland
hallett@inf.ethz.ch
[2] Dept. of Numerical Analysis of Computing Science, KTH, Stockholm, Sweden
jensl@nada.kth.se

Abstract. Evidence indicates that members of many gene families in the genome of an organism tend to have homologues both within their own genome and in the genomes of other organisms. Amongst these homologues, typically only one or a few per genome perform an analogous function in their genome. Finding subsets of these genes which show evidence of performing a common function is an important first step towards, for instance, the creation of phylogenetic trees, multiple sequence alignments and secondary structure predictions.
Given a collection of taxa $P = \{P_1, P_2, \ldots, P_k\}$ where P_i contains genes $\{p_{i,1}, p_{i,2}, \ldots, p_{i,n_i}\}$, we ask to choose one gene from each of the taxa P_i such that these chosen vertices *most agree*. We define *most agreeing* in four distinct ways: most tree-like, pairwise closest, pairwise most similar, and smallest most tree-like.
We show these problems to be computationally *hard* from almost every angle via classical, parameterized and approximation complexity theory. However, on the positive side, we give *randomized approximation* algorithms following ideas from [GGR98] for the *pairwise closest* and *pairwise most similar* variants.

1 Introduction

Given a new nucleo- or peptide sequence, the standard "first step" of any inquiry into the determination of the evolution, chemical properties, and (ultimately) function of this biomolecule is to align it against every entry in a large molecular dataset such as EMBL[S99] or SwissProt[BA]. Since properties such as *function* are extremely complex and still largely unknown, no simple search of a dataset can answer these questions directly. The standard alignment tools [AGMWL90,PL88] only return entries which show statistically significant signs of *pairwise* evolutionary relationships. The end result is that many of the returned sequences will belong to gene families other than the family of our new sequence.

There are many reasons why this is the case. We discuss three such causes below.

(1) **Domain Agreement.** Often, only a few short subsequences of one gene are homologous with other members of the gene family. These common subsequenes typically correspond to *domains*, *modules* or *motifs* that have travelled through evolution as packages. Although these subsequences are long enough and the alignments good enough as to indicate signficant similarity, the gene may perform a wildly different function.

(2) **"Long Distance Homology".** As evolutionary distances between sequences increase, it becomes increasing harder to distinguish between signficant ancestral relationships between sequences and simply noise. At extremely far evolutionary distances[1], pairwise alignments are typically between two sequences in different protein super-families. Although these protein super-families share broad macro similarities, the specific proteins in different super-families will perform extremely different functions.

(3) **Parology.** Two homologous genes are said to be *orthologous* if they evolved from a single gene existing in the genome of their lowest common ancestor taxa. Two genes are *paralogous*

⋆ Parts of this paper were submitted to SODA '00.
[1] For example, percent identity below 17% or PAM distances greater than 250.

if their lowest common ancestor can be traced back to an evolutionary event which is not a speciation. Paralogues are the result of genome level evolutionary events such as *duplications*. In essence, these events copy a contiguous strand of DNA in the genome of a taxa; any genes located along this strand are copied and proceed through evolution independently of each other. Historical reconstructions for gene lineages are typically represented as *gene trees*. The historical reconstruction of the relationships between taxa is termed a *phylogenetic* or *evolutionary tree*. The two will not necessarily agree on topology. When a gene is duplicated, one of several possibilities may occur. Firstly, it may be the case that the organism simply does not need a second copy of the gene. The gene, freed from any functional constraints in the organism, may begin to drift towards randomness, changing from a potentially active gene to a pseudogene to finally a random sequence. Secondly, as above, the organism does not require a second copy of the gene and the gene drifts towards randomness. However, after a suitable period of evolution, the gene (or more specifically, parts of the gene) may be recruited for a new function (see [SM98] for a good first treatment of how often this has happened). Thirdly, it may be the case that a second copy of the gene provides some benefit to the organism. Since it is under functional constraints, the gene is not allowed to drift towards randomness and retains an analogous function. In both of the first two cases, we are no longer interested in the resultant sequences.

In any study of evolution, chemical properties, or function, care must be taken to use sequences that are *all* pairwise homologous (all related by a common evolutionary ancestor) and that all perform an analogous function[2] in their respective genome. When such care is not taken in the selection of sequences, gene trees will not reflect the true evolutionary relationships of the species, multiple sequence alignments will not display regions of conservation and change, and predictions of secondary structure will be inaccurate [B92,BDDEHY98,F88].

We introduce the following model of the above selection problem. A collection of sets $P = \{P_1, P_2, \ldots, P_k\}$ is given where P_i corresponds to taxa i and contains the homologues $\{p_{1,i}, p_{2,i}, \ldots, p_{n_i,i}\}$ found in the genome of taxa i. The goal is to choose one gene from each of the P_i such that these genes *agree the most*. Such a subset is refered to as a *core* of the weighted k-partite graph. We introduce four distinct definitions of *most agreeing*: most tree-like, pairwise closest, pairwise most similar, and smallest most tree-like.

Most Tree Like Assuming that the taxa under study all possess exactly one gene performing an analogous function to the gene family, we arrive at the following problem:

MOST TREE LIKE IN A k-PARTITE GRAPH (CORE-TREE)
input: A complete k-partite graph $G = (P_1, P_2, \ldots, P_k, E)$, edge weights $w : E \to \mathbb{R}$.
output: A set $P' = \{p_1, p_2, \ldots, p_k\}$ where $p_i \in P_i$ such that $||D(P') - A(D(P'))||_z$ is minimized where $D(P')$ is the distance matrix formed in the obvious way from P' and $A(D(P'))$ is the closest additive approximation to $D(P')$ under the L_z norm for some $z \in \{1, 2, \ldots, \infty\}$.

That is, one vertex (one gene) is selected from each partition (each genome) such that the distance matrix formed from the pairwise comparisons of the genes is as close to additive (as close to "tree-like") as possible. The assumption behind this optimization criteria is that genes, which have a different function (hence, a significantly different underlying sequence) than the gene family, should introduce non-additivity when placed into a distance matrix consisting of genes from the gene family.

Consider point (1) above. Sequences not in the gene family will likely possess sub-regions donated from other gene families. These subregions will likely have a phylogeny much different

[2] We say *analogous function* here and not simply *function* to stress that the role a specific gene in a family plays is almost never exactly the same between organisms.

than the phylogeny of our fixed gene family. Consider point (3) above. Since paralogous genes which are not needed by the organism drift faster than genes under functional constraints and since paralogues are allowed to drift in a random direction (possibly in and out of pseudo-gene status), their sequences will likely mutate in a random direction away from the gene family, no longer following the phylogeny of the gene family. However, genes which are truly in the gene family should display (close to) "tree-like" behaviour. See Figure 1.

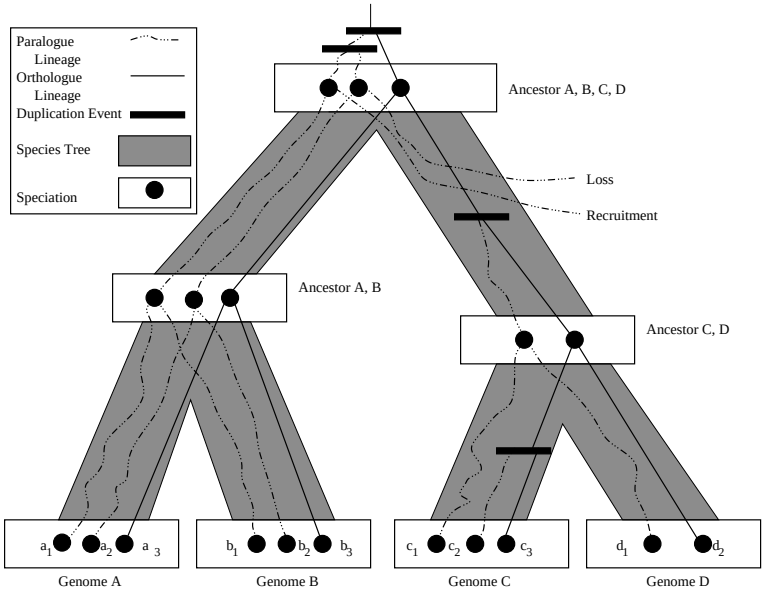

Fig. 1. The CORE-TREE PROBLEM and Paralogy. The species phylogeny for the four genomes A, B, C, D is the shaded region. Black lines represent the evolutionary history of the active orthologues whilst wavy lines protray the evolutionary history of the parologues and/or functionally inactive genes. It is assumed here that the wavy lines represent distance measurements which are (a) much larger and (b) induce distance matrices which are much further from additivity than the black lines since they are allowed to mutate quickly and in arbitrary directions.

Pairwise Closest, Pairwise Most Similar If functionally inactive genes drift quickly in a random direction through the amino acid sequence "space" and functionally active genes in our family mutate relatively slowly, then the genes performing analogous function are identifiable by being mutually more similar or closer in distance than any another homologues. Furthermore, sequences which have domains foreign to the gene family will also induce distance measures significantly greater than pairwise measurements between members of the gene family. Figure 2 graphically shows the idea here.

We arrive at our second and third notions of *most agreeing*:

MINIMUM WEIGHT CLIQUE IN A k-PARTITE GRAPHS (CORE-CLIQUE)
input: A complete k-partite graph $G = (P_1, P_2, \ldots, P_k, E)$, edge weights $w : E \to \mathbb{R}$.
output: A set $P' = \{p_1, p_2, \ldots, p_k\}$ such that $p_i \in P_i$ and $\Sigma_{1 \leq i < j \leq k} w(p_i, p_j)$ is minimum.

Note that the edges between vertices in different partitions could correspond to either (1) an estimate of the distance between the two genes, or (2) a statistical measure of similarity (eg. a maximum likelihood score). The first variant induces a minimization problem whilst the second variant induces a maximization problem. In most cases, the behaviour of either problem is the same and thus we focus attention on the former. Note also that the gene family is not

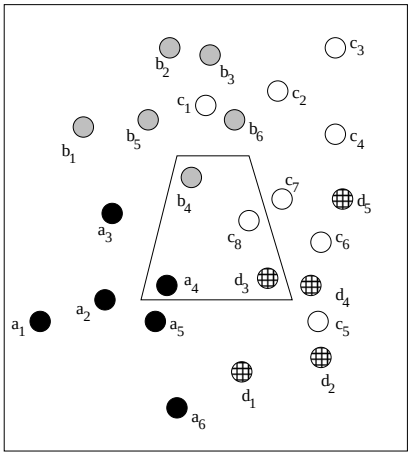

Fig. 2. A two-dimensional view of closeness. We have four genomes (black, white, grey and checkered) and we have laid out the genes in two dimensions so that topological distance is proportional to pairwise distance between the sequences. The CORE-CLIQUE problem tries to find this "core" set of mutually agreeing genes (vertices). Here we have choosen a_4, b_4, c_8, d_3.

assumed to have any sort of nice "tree-like" behavior. This problem may be particularly suited to studying microbial taxa as it is becoming clear that gene and species phylogenies are often tentative at best.

Small-Good-Core-Tree. Suppose all of the genes represented in our k-partite graph have evolved from a common ancestor through a sequence of duplications and speciations. That is, all the entries in our matrix are orthologues or paralogues with each other (point (3) above). Then, theoretically, this distance matrix could still be close to additive. Furthermore, suppose that paralogoues (presumably functionally inactive) drift much faster than orthologues (presumably functionally active). Then the orthologues should be identifiable by being members of the smallest tree in the k-partite graph, if in fact the gene and species tree agree.

SMALL TREE IN A k-PARITITE GRAPH (SMALL-CORE-TREE)
input: A complete k-partite graph $G = (P_1, P_2, \ldots, P_k, E)$, edge weights $w : E \to \mathbb{R}$.
output: A set $P' = \{p_1, p_2, \ldots, p_k\}$ such that the closest additive approximation of P' induces a tree T such that $\Sigma_{\forall e \in E_T} w(e)$ is minimum.

We do not optimize on the error between the distance matrix induced by P' and its closest additive approximation, so if the distances in G are not additive, certain degenerate conditions may occur. Note that the CORE-CLIQUE problem and the SMALL-CORE-TREE problem do not necessarily agree. It is easy to construct two matrices D and D' such that $w(D) < w(D')$ but $w(T(D)) > w(T(D'))$ where $w(D)$ is the sum of the entries in the upper triangle of the matrix and $w(T(D))$ is the weight of the edges in the tree. With this in mind, we opt to combine our notion of "close to additive" and minimum weight tree to form the following problem:

SMALL GOOD-FITTING TREE IN A k-PARTITE GRAPH (SMALL-GOOD-CORE-TREE)
input: A complete k-partite graph $G = (P_1, P_2, \ldots, P_k, E)$, edge weights $w : E \to \mathbb{R}$, $\Delta \in \mathbb{R}$.
output: A set $P' = \{p_1, p_2, \ldots, p_k\}$ such that $||A(D(P')) - D(P')||_\infty \leq \Delta$ and $\Sigma_{\forall e \in E_T} w(e)$ is minimum.

In the remainder of this paper we show that choosing cores under any of these optimization criteria is hard from the classical, parameterized and approximation complexity frameworks.

That is, the general versions of these problems are NP-complete and hard for complexity class $W[1]$ for versions of the problem when the number of partitions, the size of each partition, the maximum weight of an edge, or the overall weight of the core are parameters. We also show that all of these problems are not approximable within a polynomial function of n in polynomial time. On the positive side, we give a *randomized approximation* algorithm using ideas from [GGR98,RS96] for these last two problems. For a confidence parameter δ and a accuracy parameter ϵ, this algorithm will correctly find a core-clique of weight $opt + \epsilon\sigma \cdot k^2$ with probability $1 - \delta/2$, where *opt* is the optimal weight core clique in the input graph, k is the number of partitions and σ is the maximum difference between the weight of two edges adjacent to the same vertex. We also give a heuristic for the ORTHO-TREE and SMALL-ORTHO-TREE problem which performs very well in practice.

A Note Concerning Definitions and Previous Work in the Literature. Recently, much (due) attention has been focused on problems regarding the identification of paralogues in datasets [F88,GCMRM79,GMS96,MLZ98,MMS95,P98]. Observing that gene trees and species tree need not agree on topology when duplications and losses take place, Goodman et. al. [GCMRM79] proposed the DUPLICATION/LOSS MODEL. Here they are attempting to find the species tree which requires the fewest number of postulated events needed to rectify the observed gene trees. See also [FHKS98,FHS98] amongst others. Implicitly, this model assumes that duplication and subsequent loss events are the major cause of this disagreement. It seems almost certain that this is not the only cause (we cite point (1) above and [SM98]) and it remains unclear whether duplications and losses would even be the major cause of disagreement between gene and species tree. In [YEVB98], the authors develop a system based on BLAST, the concept of the universal tree of life, and the duplication/loss model to identify orthologues in the results of a *one-vs-all match*. Our algorithms here could be used as an important "pre-processing" step to their software as follows. Firstly, note that no matter how many duplications and losses take place, gene histories are still "tree like" even if they are not in agreement with the theoretical species tree. Therefore, our CORE algorithms will return sequences participating in the same gene tree. This will remove bad sequences such as those discussed in points (1) and (2) above. In fact, if the gene and species tree do agree (or are close in agreement), then the CORE algorithms will return the orthologues. Figure 3 provides a graphical description of these definitions and concepts. The power here is that, unlike the DUPLICATION/LOSS MODEL, we are using important distance estimates between sequences and we are placing constraints on the quality of the tree. Figure 3 provides a graphical description of these definitions and concepts.

2 Background

Definition 1 (Trees and Graphs). *A* phylogenetic tree $T = (V, E)$ *is a binary connected acyclic graph. A* leaf *in T has degree 1 and L_T is used to denote the subset of V which contain the leaves of T. For $S \subseteq T$, we let $T[S]$ represent the subtree of T induced by S. A* weighted phylogenetic tree *is a phylogenetic tree with a weight function associated with the edges, $T = (V, E, w)$ where $w : E_T \to [0, \infty)$. A complete k-partite graph is $(k+1)$-tuple $P = (P_1, P_2, \ldots, P_k, E)$ where P_i contains vertices $\{p_{i,1}, p_{i,2}, \ldots, p_{i,n_i}\}$ for some n_i where $P_i \cap P_j =$, and where E, the edge set, contains edges between every two vertices in two different partitions P_i and P_j. Weighted k-partite graphs are defined similarly. A* clique *of size t in a graph G is a set of t distinct vertices which are mutually adjacent. The weight of an edge is written $w(x, y)$ as a short hand for $w((x, y))$ for some edge (x, y).*

Definition 2 (Distance/Similarity Matrices). *A* distance matrix *D is a 0 diagonal, symmetric, nonnegative matrix, indexed by the set of taxa L_T for a phylogenetic tree T where the*

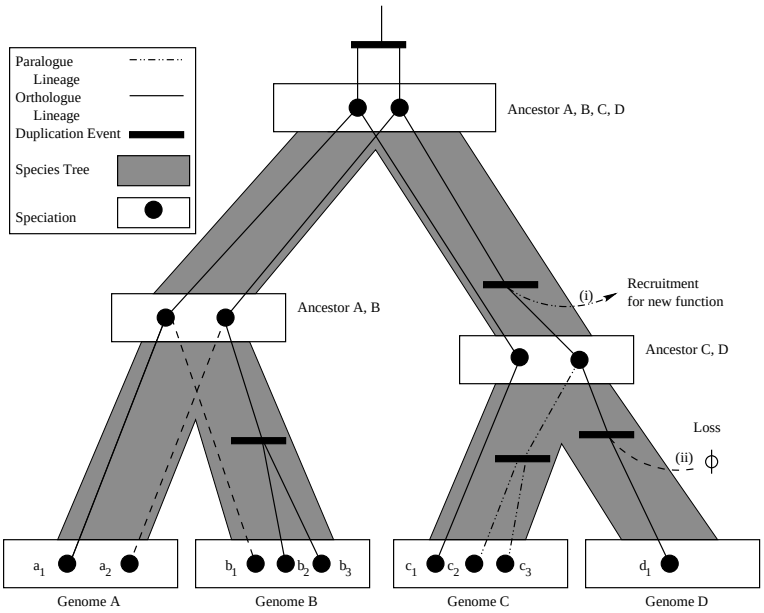

Fig. 3. The basic concepts. The extant genomes here are $A = \{a_1, a_2\}$, $B = \{b_1, b_2, b_3\}$, $C = \{c_1, c_2, c_3\}$, and $D = \{d_1\}$. The solid lines represent the evolutionary history of the functionally active genes whilst the dotted lines represent the history of the functionally inactive ones. The duplication directly before the ancestor C, D created a new gene that was recruited for a new function. The duplication below this vertex created a gene that was loss (either through drift or through a deletion event). Notice here that the gene and species tree do not agree – $((A, C), (B, D))$ and $((A, B), (C, D))$ resp. The functionally active genes in this configuration are $\{a_1, b_2, b_3, c_1, d_2\}$. Note that genome B has two such genes due to the very recent duplication event.

entry D_{ij} is the distance (an estimated distance) between taxa i and taxa j. An $n \times n$ distance matrix D is additive*, if there exists a weighted phylogenetic tree T with n leaves such that entry D_{ij} equals to the sum of the edge weights in the tree along the path connecting i and j. A* similarity matrix *S is the same as a distance matrix except that diagonal elements have value ∞ and entry S_{ij} is a similarity score between taxa i and j.*

Theorem 1 ([B71]). *A matrix D is additive if and only if for all i, j, k, l (not necessarily distinct), the maximum of $D_{ij} + D_{kl}, D_{ik} + D_{jl}, D_{il} + D_{jk}$ is not unique. The edge weighted tree (with positive weights on internal edges and non-negative weights on leaf edges) representing the additive distance matrix is unique among the trees without vertices of degree two.*

Definition 3 (Error Measurements). *The L_k norm between distance matrices D and D', written $||D - D'||_k$, is defined as $||D - D'||_k = (\Sigma_{i<j} \left(\left| D_{ij} - D'_{ij} \right| \right)^k)^{\frac{1}{k}}$ for $k \geq 1$. For $k = \infty$, the L_∞ norm is defined as $||D - D'||_\infty = max_{i<j} \left| D_{ij} - D'_{ij} \right|$*

Definition 4 (Approximation Ratios). *An approximation algorithm is said to achieve an* approximation ratio *of α for a maximization problem Π if for each input x, it computes a solution y of cost at least OPT/α, where OPT is the cost of the optimum. For a minimization problem, the algorithm must return a solution y of cost at most $\alpha \cdot OPT$. Note that $\alpha \geq 1$.*

Theorem 2 (Hoeffding Bound [H63]). *If X be the sum of n independent and* bounded *random variables $X_i \in [a_i, b_i]$ and let $\bar{X} = X/n$, then for $t > 0$,*

$$Pr[\bar{X} - E[\bar{X}] > t] \leq exp\left(-\frac{2n^2t^2}{\Sigma_{i=1}^{n}(b_i - a_i)^2} \right)$$

or, equivalently,

$$Pr[X - E[X] \geq t] \leq exp\left(-\frac{2t^2}{\Sigma_{i=1}^{n}(b_i - a_i)^2}\right)$$

2.1 Parameterized Complexity

We refer the reader to [DF99] for a complete description of parameterized complexity.

Parameterized computational complexity, introduced by Downey and Fellows [DF99], is founded on the observation that the overwhelming majority of problems take as input two or more parameters. They are concerned with languages $L \subseteq \Sigma^* \times \Sigma^*$ and if $\langle x, k\rangle$ is in a parameterized language L, we call k the parameter. In the interests of readability and with no loss of generality to the theory, we assume that the parameter k has domain $\mathbb{N}$; that is, $L \subseteq \Sigma^* \times \mathbb{N}$. For fixed k, we call $L_k = \{x | \langle x, k\rangle \in L\}$ the k-th *slice* of L.

The primary intention is to study languages that are *tractable* by this "slice". This theory was motivated by the observation that for many problems only a small range of values for some input parameters capture most instances arising in practice.

Definition 5 (The Good - FPT). *For a parameterized language L, we say that L is* (uniformly) fixed parameter tractable *(FPT) if there exists a constant α and an algorithm Φ such that Φ decides if $\langle x, k\rangle \in L$ in time $f(k)|x|^{\alpha}$ where $f : \mathbb{N} \to \mathbb{N}$ is an arbitrary function.*

Although f may be exponential (or worse), such an algorithm for recognizing a language may provide a perfectly feasible (exact) solution to the problem in practice. However, many languages seemingly do not admit such behaviour and require time $\Omega(n^{f(k)})$ with $f(k) \to \infty$ for a size n problem with a solution set of size k. The notion of the *bad* is intuitively associated with trying to beat the naive algorithms of trying all $\binom{n}{k} = \Theta(n^k)$ subsets or using k dimensional dynamic programming. The Clique Problem, which asks if there is a set of k mutually adjacent vertices in the graph of size n, is one such example with the best known algorithms using time $\Theta(n^{\theta \cdot k})$, where θ is the constant for matrix multiplication. Completeness frameworks typically consist of a notion of a set of languages at least as hard as all other languages in the class and a notion of complexity preserving reduction. Theorem 3 and Definition 6 provide these two concepts resp.:

Theorem 3 (The Bad - $W[1]$ completeness, citeDF99). *k-Clique, parameterized by the clique set size k, is complete for complexity class $W[1]$.*

Definition 6 (Parameterized many:1 reductions). *We say that L reduces to L' by a standard parameterized m–reduction if there is an algorithm Φ which transforms $\langle x, k\rangle$ into $\langle x', g(k)\rangle$ in time $f(k)|x|^{\alpha}$, where $f, g : \mathbb{N} \to \mathbb{N}$ are arbitrary functions and α is a constant independent of k, so that $\langle x, k\rangle \in L$ if and only if $\langle x', g(k)\rangle \in L'$.*

It follows that k-Clique can not be solved in FPT time, unless $W[1] = FPT$. This seems rather unlikely and there now exists a volume of evidence supporting this conjecture. A problem that is *hard* for $W[1]$ is at least as hard as all problems in $W[1]$.

3 Complexity Results

3.1 Core-Clique

We begin with an analysis of the Core-Clique problem.

CORE-CLIQUE
input: A complete k-partite graph $G = (P_1, P_2, \ldots, P_k, E)$, edge weights $w : E \to \mathbb{R}$, $r \in \mathbb{R}$.
(decision) question: Does there exist a set $P' = \{p_1, p_2, \ldots, p_k\}$ such that $p_i \in P_i$ and $\Sigma_{1 \leq i < j \leq k} w(p_i, p_j) \leq r$?
(optimization) output: $P' = \{p_1, p_2, \ldots, p_k\}$ such that P' minimizes $\Sigma_{1 \leq i < j \leq k} w(p_i, p_j)$.

We restrict our attention to the L_∞ norm throughout the following analysis, but note that our reductions also work for the other norms. The decision version of this problem takes as input a parameter $r \in \mathbb{R}$ and answers "yes" iff the core-clique has weight $\leq r$. Theorem 4 below states that even when the number of candidate genes per genome bounded by 3, an extremely simple weighting function is used, and a bound of 0 is placed on the size of the core-clique, the problem remains NP-complete. Theorem 5 states that a modified (easier to approximate) version of CORE-CLIQUE cannot be approximated within any function of n (the number of vertices of the input graph) in polynomial time. Both these theorems follows easily from the following lemma.

Lemma 1. *Let $f(n)$ be a function such that $f(n) > 0$ for all $n \geq 1$, then* CORE-CLIQUE *restricted to partitions of size* 3 *and with a weighting function w which assigns an edge either* 0 *or $f(n)$, and $r = 0$ is NP-complete, where n is the size of the input graph.*

Proof. The problem is in NP. To show hardness, we reduce from 3SAT.

3SAT
input: A formula Φ in 3-CNF over a set of variables $X = \{x_1, x_2, \ldots, x_t\}$.
question: Is there a truth assignment to X such that each clause of Φ has at least one literal is true?

Let $X = \{x_1, x_2, \ldots, x_t\}$ be the set of variables and $C = \{C_1, C_2, \ldots, C_k\}$ be the set of clauses of an arbitrary instance of this problem. To construct an instance of the CORE-CLIQUE problem (G, w, r), we create k partitions $P_1, P_2, \ldots, P_k$ and associate P_i with clause C_i. The 3 vertices in P_i are labeled by the literals in C_i. The weight of an edge between two vertices in different partitions corresponding to two negated literals x_j and $\bar{x}_j$ is $f(n)$. Otherwise, the weight is 0.
Claim G has a weight 0 *core-clique if and only if Φ is satisfiable.*
($\Rightarrow$) Let $p^1, p^2, \ldots, p^k$ be the set of vertices which induce a core-clique of weight 0. Now there can be no weight $f(n)$ edges between any p^i and p^j which implies that it is never the case that p^i is some literal x whilst p^j is the negated literal $\bar{x}$. Hence, we may set the literal p^i to be true. Since we may do this for all k of the partitions, we have a truth assignment for Φ with at least one true literal in each clause.

($\Leftarrow$) Let $\mathcal{T} : X \to \{true, false\}$ be a truth assignment to Φ such that at least one literal x in each clause C_i is true. Consider any two distinct such literals x_i and x_j which are true in clauses C_i and C_j. Then the vertex lapelled x_i in P_i and the vertex lapelled x_j in P_j have no weight $f(n)$ edge between them, since $\mathcal{T}$ is a satisfying assignment for Φ and there is an edge of weight $f(n)$ only if two literals are negations of each other. Hence, we may place x_i and x_j in the core-clique.

Theorem 4. CORE-CLIQUE *restricted to partitions of size* 3 *and with a weighting function w which assigns an edge either* 0 *or* 1, *and $r = 0$ is NP-complete.*

No minimization problem for which it is NP-complete to distinguish between instances with 0 minimum cost and instances with cost $c > 0$ can be approximated within any ratio

in polynomial time. Since this comment applies to the CORE-CLIQUE problem, we formulate a slightly modified version of the optimization form (MODIFIED-CORE-CLIQUE) of the problem which asks for the P' which minimizes $1 + \Sigma_{1 \leq i < j \leq k} w(p_i, p_j)$, for which non-trivial non-approximability results can be proved.

Theorem 5. *If $P \neq NP$, then* MODIFIED-CORE-CLIQUE *is not approximable within any function of n in polynomial time, where n is the size of the input graph.*

Proof. Assume that MODIFIED CORE-CLIQUE can be approximated in polynomial time approximated to within a function $g(n)$. It follows immediately that $g(n) \geq 1$ for all $n \geq 1$. By Lemma 1, it is NP-hard to distinguish between instances of MODIFIED CORE-CLIQUE with a minimum of 1 and those with a minimum of $1 + g(n)$. However, using the assumed approximation algorithm it is possible to distinguish between such instances. From this contradiction the theorem follows.

Next we examine the CORE-CLIQUE problem from the perspective of parameterized complexity (see § 2 and [DF99]). The main principle here is that, although the general form of the problem is NP-complete, our reduction does not disclose exactly where the source of intractability lies. We see at least the following four possible parameterizations of the problem:
(1) $m = max_{\forall i}|P_i|$, the maximum size of a partition,
(2) k, the number of partitions,
(3) r, the total weight of the core-tree, and
(4) ω, the maximum weight of a distance between two leaves.
Note, Theorem 4 shows that any subset of parameters 1, 3 and 4 are not enough as the problem remains NP-complete. Our next theorem rules out the possibility of an FPT algorithm for any subset of parameters 2, 3, and 4.

Theorem 6. $2,3,4$-CORE-CLIQUE *is hard for* $W[1]$.

Proof. Let $(C = (V, E), K)$ be an instance of the K-CLIQUE PROBLEM. We construct an instance of the CORE-CLIQUE problem $(G = (P_1, P_2, \ldots, P_k, E), w, r)$, where r, ω, and k are functions depending only on K and show that (C, K) is a "yes" instance if and only if (G, w, r) is a "yes" instance.

Let the vertices in V_C be labeled by $1, 2, \ldots, |V_C| = m$. Let $r = \binom{K}{2}$. We create partitions $P_1, P_2, \ldots, P_{K=k}$ and include vertices labeled $p_{i,j}$ for $1 \leq j \leq m$ in partition P_i. We place an edge between all vertices in G which are not in the same partition: for all i, j, $1 \leq i < j \leq k$, and for all q, q', $1 \leq q < q' \leq m$, $(p_{i,q}, p_{j,q'}) \in E_G$. If $(u, v) \notin E_C$, then $w(p_{i,u}, p_{j,v}) = c$ for all $1 \leq i < j \leq k$. c is an arbitrarily large constant at least as big as $\binom{K}{2} + 1$. If $(u, v) \in E_C$, then $w(p_{i,u}, p_{j,v}) = 1$ for all $1 \leq i < j \leq k$. For all edges of the form $(p_{i,u}, p_{j,u}) \in E_G$, let $w(p_{i,u}, p_{j,u}) = c$.
($\Rightarrow$) Let $V' = \{v_1, v_2, \ldots, v_K\}$ be the clique set in C. By the construction, there must exist edges in G of the form (p_{i,v_i}, p_{j,v_j}) with weight 1. Hence, the core-clique consisting of $\{p_{1,v_1}, p_{2,v_2}, \ldots, p_{K=k,v_K}\}$ in G has weight $\binom{K}{2}$.
($\Leftarrow$) Let P' be the core-clique consisting of vertices $\{p_1, p_2, \ldots, p_k\}$ with weight bound $r = \binom{K}{2}$, where $p_i = p_{i,v}$ is a vertex in P_i. Since edges have either weight 1 or weight $c > \binom{K}{2}$ in G, all edges induced by P' must have weight 1. Therefore, by the construction, all edges $(p_{i,v}, p_{j,v'})$ are contained in E_C. Hence, these vertices form a clique in C.

Observation 1 $1,2$-CORE-CLIQUE *is fixed parameter tractable with an algorithm running in time* $O(n^k)$.

Proof. Simply try all $O(n^k)$ valid sets of k vertices.

Theorem 4 shows that the problem remains hard for partition size 3 with constant edge weight functions and a constant bound on the core-clique. Our next theorem shows that restricted to partition size 2 and constant edge weight functions it still stays hard.

We reduce from the MAXIMUM 2SAT problem:

MAXIMUM 2-SATISFIABILITY[GareyJ79]
input: A formula Φ in CNF over a set of variables $X = \{x_1, x_2, \ldots, x_m\}$ such that each of the l clauses $c \in \Phi$, $|c| = 2$, $K \in \mathbb{Z}$.
question: Is there a truth assignment for Φ that simultaneously satisfies at least K of the clauses?

Theorem 7. *1, 4-CORE-CLIQUE is NP-complete even when the number of vertices in each partition is at most 2 and the edges are assigned a weight of either 0 or 1.*

Proof. Clearly, the problem is in NP.

Let (Φ, K), where Φ consists of clauses $C_1, C_2, \ldots, C_l$ be an instance of the MAXIMUM 2-SATISFIABILITY problem. We construct an instance $(G = (P_1, P_2, \ldots, P_k, E), w, r)$ of the CORE-CLIQUE problem as follows: for each variable $x_i \in X$, we construct a partition P_i consisting of two vertices labelled p_i and $\bar{p}_i$ (corresponding to a postive and negative truth assignment to x_i). Hence, $k = |X| = m$ and $max_{\forall i}|P_i| = 2$. For each clause $C_i \in \Phi$, where C_i consists of literals (x^u, x^v), where x^u is either x_u or $\bar{x}_u$ and x^v is either x_v or $\bar{x}_v$, we assign an edge of weight 1 between the two vertices of P_u and P_v corresponding to $\bar{x^u}$ and $\bar{x^v}$, the negated literals. All other edges have weight 0. Let $r = l - K$.

Claim. (G, w, r) is a "yes" instance of the CORE-CLIQUE problem if and only if (Φ, K) is a "yes" instance of the MAXIMUM 2-SATISFIABILITY problem.

($\Rightarrow$) Let $P' = \{p^1, p^2, \ldots, p^k\}$ be the core-clique in G which has weight $\leq r = l - K$. Since edges of weight 1 only occur between partitions P_i and P_j where x^i and x^j appear together in a clause of Φ, we have exactly l edges of weight 1 in G and P' must be such that the core-clique has at most $l - K = r$ of these edges. This implies the existence of at least K distinct pairs of partitions (P_i, P_j) such that (p^i, p^j) has a weight 0 edge. By the construction, p^i (resp. p^j) is either p_i or $\bar{p}_i$ (resp. p_j or $\bar{p}_j$) and corresponds to assigning literal x_i (x_j) true or false. Since no edge of weight 1 exists between p^i and p^j, the corresponding clause in Φ is satisfied. Therefore, there are at least K clauses in Φ which are satisfied.

($\Leftarrow$) Let $\mathcal{T} : X \rightarrow \{true, false\}$ be a truth assignment to Φ satisfying at least K clauses $\{C^1, C^2, \ldots, C^K\}$. By the construction, for each C^i consisting of literals (x^u, x^v), the partitions P_u and P_v contain one edge between them of weight 1. C^i is satisfied so $T(x^u) \cup T(x^v)$ is not false. If $T(x^u) = true$ (resp. $T(x^v) = true$), we place p_u (p_v) in the core-clique. Otherwise, we place $\bar{p}_u$ ($\bar{p}_v$) in the core-clique. The edge between these two parititions has weight 0. Since there are at least K such C_i's and there are exactly l edges of weight 1 only appearing between partitions with variables simultaneously in a clause of Φ, the overall weight of the core-clique is less than or equal to $l - K = r$.

3.2 Most Tree Like

BEST TREE IN A k-PARTITE GRAPH (CORE-TREE)
input: A complete k-partite graph $G = (P_1, P_2, \ldots, P_k, E)$, edge weights $w : E \rightarrow \mathbb{R}$.
output: A set $P' = \{p_1, p_2, \ldots, p_k\}$ where $p_i \in P_i$ such that $||D(P') - A(D(P'))||_\infty$ is minimized where D is the distance matrix formed in the obvious way from P' and $A(D(P'))$ is the closest additive approximation to D under the L_∞ norm.

Clearly, the decision version of the CORE-TREE problem, which asks if there is a P' such that $||D(P') - A(D(P'))||_\infty \leq \Delta$ for input parameter $\Delta \in \mathbb{R}$, is NP-complete since NUMERICAL TAXONOMY [ABFNPT96] [3] is simply a restricted version (specifically, all partitions having size 1) of it. We begin our analysis with a sub-version of the problem where we ask if there exists a choice of one leaf from each partition in the input graph that induces an additive tree. Furthermore, we are given the unweighted topology of the tree, so the problem reduces to just choosing one vertex per partition so that the pairwise distances fit to the tree. This problem, when each partition just has a single vertex, is not NP-complete [F88].

EXACT TREE IN A k-PARTITE GRAPH (EXACT-CORE-TREE)

input: As with CORE-TREE but also an unweighted leaf-labeled tree T with each leaf receiving a distinct label from $\{P_1, P_2, \ldots, P_k\}$.

question: Does there exist a set $P' = \{p_1, p_2, \ldots, p_k\}$ where $p_i \in P_i$ such that $D(P')$ is additive, where $D(P')$ is the distance matrix formed from P', and such that the corresponding tree $T(D(P'))$ is isomorphic to T and for $u \in T(D(P'))$, $u \in P_i$, the corresponding leaf in T has label P_i.

Again, we analyze this problem from the perspective of parameterized complexity. Our parameters remain the same: (1) $m = max_{\forall i}|P_i|$, the maximum size of a partition, (2) k, the number of partitions, (3) r, the total weight of the core-tree, and (4) ω, the maximum weight of a distance between two leaves. Our first theorem shows that no FPT algorithms are possible for any subset of parameters 2, 3, or 4, unless $W[1] = FPT$.

Theorem 8. $2, 3, 4$-EXACT-CORE-TREE *is hard for* $W[1]$.

Proof. Given an instance of the K-CLIQUE PROBLEM $(C = (V, E), K)$, we create an instance of the $2, 3, 4$-EXACT-CORE-TREE problem (G, T) and show that (C, K) is a "yes" instance if and only if (G, w, T) is a "yes" instance.

We construct $K + 4 (= k)$ partitions $\{A, B, C, D, P_1, P_2, \ldots, P_K\}$. Partition A contains one vertex a, B contains b, C contains c, and D contains d. Each partition P_i contains $|V_C| = m$ vertices labeled $p_{i,1}, p_{i,2}, \ldots, p_{i,m}$. Our tree T is created as in Figure 4: the caterpillar with (A, B) and (C, D) as its "head" and "tail". That is, our tree has internal vertices $\{h, t, n_1, \ldots, n_K\}$ with edges $\{(h, A), (h, B), (t, C), (t, D), (h, n_1), (t, n_K)\}$ and $\{(n_i, n_{i+1}) : 1 \leq i < K\}$.

Let $D_{a,b} = D_{c,d} = 2$, $D_{a,c} = D_{a,d} = D_{b,c} = D_{b,d} = 4 + (K - 1)$. Let $D_{x,p_{i,j}} = 2 + i$ for $x = \{a, b\}$, $1 \leq i \leq K$ and $1 \leq j \leq m$. Let $D_{y,p_{i,j}} = 2 + (K - i + 1)$ for $y = \{c, d\}$, $1 \leq i \leq K$ and $1 \leq j \leq m$. Let $D_{p_{i,j},p_{i',j}} = 3K + 10$ for all $1 \leq i \neq i' \leq K$ and $1 \leq j \leq m$. If $(u, v) \notin E_C$, then $D_{p_{i,u},p_{i',v}} = 3K + 10$ for all $1 \leq i \neq i' \leq K$. If $(u, v) \in E_C$, then for $1 \leq i < j \leq K$, $D_{p_{i,u},p_{j,v}} = 2 + j - i$.

($\Rightarrow$) Let $V' = \{v_1, v_2, \ldots, v_K\}$ where $v_i \in V_C$ a clique in C. We show how to choose one vertex from each of the P_i in G such that the distance matrix formed from these vertices alongside with a, b, c and d is additive. Note that we must choose a, b, c and d, and that the distance matrix these four vertices induce is additive (see Theorem 1) and agrees with the topology T.

Now consider the set of vertices $\{p_{1,v_1}, p_{2,v_2}, \ldots, p_{K,v_K}\} = P'$ in G. From the construction, $D_{p_{i,v_i},p_{j,v_j}} = 2 + j - i$ as any two distinct vertices p_{i,v_i}, p_{j,v_j} from this set are mutually adjacent. We must show how weights can be applied to the edges of T such that the distances in T between p_{i,v_i} and p_{j,v_j}, $d(p_{i,v_i}, p_{j,v_j})$ are equal to the entries $D_{p_{i,v_i},p_{j,v_j}}$. This can be accomplished by assigning 1 to every edge on the path between p_{i,v_i} and p_{j,v_j} in T. It is easy to verify that $d_T(x, p_{i,v_i}) = D_{x,p_{i,v_i}}$, for $x \in \{a, b, c, d\}$ and that the matrix can be realized as a tree.

[3] NUMERICAL TAXONOMY. **input:** An $n \times n$ distance matrix D, a bound $\Delta \in \mathbb{R}$. **question:** Is $||A(D) - D||_\infty \leq \Delta$?

($\Leftarrow$) Let $P' = \{a, b, c, d, p_1, p_2, \ldots, p_K\}$ be the set of vertices from G which induces a tree with topology T. By Theorem 1, the underlying distance matrix D is additive. For a leaf vertex x, let $n(x)$ be the unique neighbour of x in T. Focus on the four vertices $\{a, b, c, d\}$. By Theorem 1, the edge weights in this subtree must be 1 for edges of the form $(x, n(x))$ where $x \in \{a, b, c, d\}$. The weight of the path between (a, b) and (c, d) receives weight $2 + (K - 1)$. We now analyze the "choice" of vertices $\{p_1, p_2, \ldots, p_K\}$.

Claim (No Fit). P' does not contain two vertices $p_{i,j}$ and $p_{i',j}$, $i \neq i'$.

(By contradiction) Suppose there exist $p_{i,j}, p_{i',j} \in P'$ simultaneously (w.l.o.g. $i < i'$). Then, by the construction, $D_{p_{i,j},a} = 2 + i$, $D_{p_{i,j},c} = 2 + (K - i + 1)$, $D_{a,b} = 2$ and $D_{p_{i,j},p_{i',j}} = 3K + 10$. Focus on the quartet formed by $\{a, b, p_{i,j}, c\}$. It is easy to verify that the edge $(p_{i,j}, n(p_{i,j}))$ must have weight 1. Furthermore, the path from vertex (AB) to $n(p_{i,j})$ must have total weight i and the path from vertex (CD) to $n(p_{i,j})$ must have weight $K - i + 1$. The same argument holds for the edge weights in the quartet $\{a, b, p_{i',j}, c\}$, that is, the edge weight of $(p_{i',j}, n(p_{i',j}))$ is also 1. Allowing $n(x)$ to denote the unique neighbor of a leaf vertex x in T, it is easy to verify that the weight of the path from $n(p_{i,j})$ to $n(p_{i',j})$ must be $i' - i$. Since $i' - i + 2 < 3K + 10$, we reach a contradiction since we can not assign edge weights to T so that they agree with the distance matrix induced by $\{a, b, c, d, p_{i,j}, p_{i',j}\}$. Hence, by Theorem 1, this matrix is not additive.

Claim. P' does not contain two vertices $p_{i,j}$ and $p_{i',j'}$, $i < i'$, $j \neq j$, such that $(v_j, v_{j'}) \notin E_C$.

This claim can be proved in the same way as Claim *No Fit* above. Simply note we assigned $D_{p_{i,j},p_{i',j'}}$ to be $3K + 10$ when $(v_j, v_{j'}) \notin E_C$.

The previous two claims establish the fact that we must include K distinct vertices in G which correspond to pairwise adjacent vertices in C. Hardness for $W[1]$ follows from the fact that our construction required only $K + 4$ partitions, all edge weights are a function only of K and the overall weight of the clique-tree is also a function only of K.

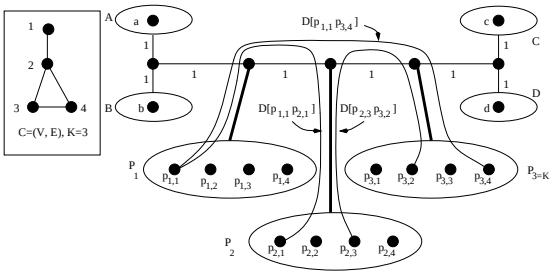

Fig. 4. Construction for the $2, 3, 4$-Exact-Core-Tree.

Our second theorem shows that this problem is NP-complete even when the number of candidate homologous genes per genome is at most 3.

Theorem 9. 1-Exact-Core-Tree *restricted to partitions of size* 3 *is* NP*-complete.*

Proof. We reduce for 3SAT. Let Φ be a formula in 3CNF form over variables $X = \{x_1, x_2, \ldots, x_m\}$ and clauses $C_1, C_2, \ldots, C_k$. We construct an instance of the Exact-Core-Tree (G, w, T) as follows. Let there be $k + 4$ partitions in G $(P_1, P_2, \ldots, P_k, A, B, C, D)$ where A contains the single vertex a, B contains b, C contains C, and D contains d. P_i contains three vertices

$p_{i,1}, p_{i,2}, p_{i,3}$ associated with the three literals in clause C_i of Φ. Our topology T is again the catepillar from Theorem 8: $(((((A,B),P_1),P_2),\ldots,P_k),(C,D))$. Let $D_{a,b} = D_{c,d} = 2$, $D_{a,c} = D_{a,d} = D_{b,c} = D_{b,d} = 4 + (k-1)$. Let $D_{x,p_{i,j}} = 2 + i$ for $x = \{a,b\}$, $1 \leq i \leq k$ and $1 \leq j \leq 3$. Let $D_{y,p_{i,j}} = 2 + (k - i + 1)$ for $y = \{c,d\}$, $1 \leq i \leq k$ and $1 \leq j \leq 3$. For $p_{i,s}$ and $p_{j,t}$, where $i \neq j$ and literal s in Φ is the negation of literal t, let $D_{p_{i,s},p_{j,t}} = 3k + 10$. When s is not the negation of literal t, let $D_{p_{i,s},p_{j,t}} = 2 + j - i$.

Claim. (G, w, T) contains an additive core-tree with topology T if and only if Φ is satisfiable.

($\Leftarrow$) Let $\mathcal{T} : X \rightarrow \{true, false\}$ be a truth assignment to Φ such that at least one literal x in each clause C_i is true. We show how to choose one vertex from each P_i in G such that the distance matrix formed from these choices alongside with a, b, c, and d are additive. Note that we must choose a, b, c and d and they are additive with a topology in agreement with T (see also Theorem 8 and Figure 4). Let $x^1, x^2, \ldots, x^k$, $x^i \in C_i$ be true literals in the clauses of Φ. Since all such literals are true, it is never the case that $x^i = \bar{x^j}$. By the construction, $D_{p_{i,x^i},p_{j,x^j}} = 2 + j - i$.

Consider the set $P' = \{p_{1,x^1}, p_{2,x^2}, \ldots, p_{k,x^k}$. We need only show how to apply edge weights to T so that the distances in T between p_{i,x^i} and p_{j,x^j} equal the entries in the distance matrix. This can be accomplished by assinging weight 1 to every edge on the path between p_{i,x^i} and p_{j,x^j}. It is easy to verify the tree distances agree with the distance matrix.

($\Rightarrow$) Let $P' = \{a, b, c, d, p^1, p^2, \ldots, p^k\}$ be the set of vertices in G which induces a tree with topology T. By Theorem 1, the underlying distance matrix D is additive. Focus on the four vertices $\{a, b, c, d\}$. By Theorem 1, the edge weights in this subtree must be 1 for edges of the form $(x, parent(x))$ where $x \in \{a, b, c, d\}$. The weight of the path between (a, b) and (c, d) receives a weight $2 + (k - 1)$. We now analyze the "choice" of vertices $\{p^1, p^2, \ldots, p^k\}$.

Claim. P' does not contain two vertices $p_{i,j}$ and $p_{i',j}$, $i \neq i'$.

(By contradiction) Suppose there exist $p_{i,j}, p_{i',j} \in P'$ simultaneously (w.l.o.g. $i < i'$). Then, by the construction, $D_{p_{i,j},a} = 2 + i$, $D_{p_{i,j},c} = 2 + (k - i + 1)$, $D_{a,b} = 2$ and $D_{p_{i,j},p_{i',j}} = 3k + 10$. Focus on the quartet formed by $\{a, b, p_{i,j}, c\}$. It is easy to verify that the edge $(p_{i,j}, parent(p_{i,j}))$ must have weight 1. Furthermore, the path from vertex (AB) to $parent(p_{i,j})$ must have total weight i and the path from vertex (CD) to $parent(p_{i,j})$ must have weight $k - i + 1$. The same argument holds for the edge weights in the quartet $\{a, b, p_{i',j}, c\}$, that is, the edge weight of $(p_{i',j}, parent(p_{i',j}))$ is also 1. It is easy to verify that the weight of the path from $parent(p_{i,j})$ to $parent(p_{i',j})$ must be $i' - i$. Since $i' - i + 2 < 3k + 10$, we reach a contradiction since we can not assign edge weights to T so that they agree with the distance matrix induced by $\{a, b, c, d, p_{i,j}, p_{i',j}\}$. Hence, by Theorem 1, this matrix is not additive.

Claim. P' does not contain two vertices $p_{i,j}$ and $p_{i',j'}$, $i < i'$, $j \neq j$, such that $x_j = \bar{x_{j'}}$ where $x_j \in C_i$ and $x_{j'} \in C_j$.

This claim can be proved in the same way as Claim 3.2 above. Simply note we assigned $D_{p_{i,j},p_{i',j'}}$ to be $3k + 10$ when x_j and $x_{j'}$ appear in two distinct clauses as complements of each other.

The previous two claims establish the fact that we must include k distinct vertices from G which correspond to conflict free choices for true literals in Φ.

Parameterizing on both the number of partitions k and the size of each partition m leads to a trivial FPT algorithm for 1, 2-Core-Tree with a running time of $O(m^k)$.

Observation 2 1, 2-Core-Tree *is FPT and solvable in time* $O(n^k)$.

Consider the relaxation of EXACT-CORE-TREE to the optimization version which asks for the core-set P' which best fits to the topology T and we modify this optimization criteria so that it is always > 0, we can prove the following non-approximation results via Theorem 9:

Theorem 10. *The always positive, optimization version of* EXACT-CORE-TREE *is not approximable within any function of n in polynomial time, where n is the size of the graph G, unless $P = NP$.*

Proof. Similar to Theorem 5.

3.3 A Heuristic for the Core-Tree Problem

Given the complexity results of Theorems 8 and 9, there does not exist polynomial or FPT algorithms for this problem even for the very restricted case when the topology of the species tree is known and at least one of the core-sets induce an additive distance matrix, unless extremely unlikely complexity collapses occur. Hence, we must be satisfied at this stage to accept a heuristic solution. The algorithm given here combines the randomization techniques used in CORE-CLIQUE with the NEIGHBOUR JOINING technique (NJ) [SN87]. The NJ method will reconstruct the correct topology if the amount of non-additivity in the distance matrix does not exceed half the length of the smallest edge.

Theorem 11 ([A98]). NJ *returns the correct topology for a phylogenetic tree when $||A(D) - D||_\infty < \frac{x}{2}$ where x is the smallest edge in $A(D)$.*

Our algorithm computes all possible NJ trees from a randomly choosen small (S where $|S| = \Theta(log\ k)$) set of partitions. Each tree is scored via the least squares (L_2 norm) fit between the distance matrix and the NJ tree. This "kernel" set is extended greedly partition by partition, again computing the optimal error via the least squares algorithm for the new NJ tree.

CORE-TREE ALGORITHM

We repeat the following $O(n^2)$ times:

1. Randomly choose a sample set $S = \{s_1, s_2, \ldots, s_{|k|}\}$, $S \subset \{1 \ldots k\}$ of $\Theta(log\ k)$ distinct partitions.
2. Compute $LS(NJ(D(p_{s_1}, p_{s_2}, \ldots, p_{s_{|}S|})), D(p_{s_1}, p_{s_2}, \ldots, p_{s_{|}S|}))$ for all $p_{s_i} \in P_i$ where $D(S)$ is the distance matrix induced by vertex set S, $NJ(D)$ is the tree topology returned by the NEIGHBOUR JOINING algorithm [SN87] on distance matrix D, and $LS(T, D)$ is an algorithm which returns the optimal fit of the distance matrix D to the tree topology T under the L_2 norm. Let $\mathcal{T} = \{T_1, T_2, \ldots, T_{\Pi_{i=1}^{S}|P_{s_i}|}\}$.
3. **for** each $T_i \in \mathcal{T}$ **do**
   for P_i, $i \notin S$, **do**
   Let $T_i = T_i \cup v$ where $v \in P_i$ minimizes
   $LS(NJ(D(V_{T_i} \cup v)), D(V_{T_i} \cup v))$
   $S = S \cup i$
   do do

In practice, we compute the optimal core-tree exhaustively when the input graph is small enough.

4 A Randomized Approximation Algorithm for the CORE-CLIQUE Problem

Following [GGR98], we will now give a randomized approximation algorithm for the CORE-CLIQUE problem. The algorithm runs in linear time if each P_i has size bounded by a constant m, and polynomial time in the general case. Let $\sigma(G, w)$ denote the maximum difference between the weights of two edges adjacent to a vertex v, over all vertices v of G and its adjacent edges.

Theorem 12. *For any $\epsilon, \delta \in (0, 1)$, there is a randomized algorithm for the* CORE-CLIQUE *problem that for a given instance G, w with probability $\geq 1 - \delta$ in polynomial time finds a solution of cost $\leq c^* + \epsilon\sigma(G, w)k^2$, where c^* is the cost of the minimum cost core-clique.*

Consider a given CORE-CLIQUE instance G, w and let $\sigma = \sigma(G, w)$. Let ϵ, our *distance parameter*, be such that $0 < \epsilon < 1$ and δ, our *confidence parameter*, be such that $0 < \delta < 1$. We use $[k]$ to denote the set $\{1, 2, \ldots, k\}$.

Let $l = \lceil 8/\epsilon \rceil$ and $t = \Theta(\frac{1}{\epsilon^2} \log \frac{1}{\epsilon\delta})$. Consider a partition of $[k]$ into l sets $A_1, \ldots, A_l$ of approximately equal size. Let $V_j = \cup_{i \in A_j} P_i$ and $W_j = V(G) \setminus V_j$. For $U = U_1, \ldots, U_l$ where $U_j \subseteq [k] \setminus A_j$, let $X(U_j)$ be the family of all $X \subseteq W_j$ such that $|X \cap P_i| = 1$ for all $i \in U_j$ and $X \cap P_i = \emptyset$ for $i \notin U_j$, and let $X(U) = \{(X_1, \ldots, X_l) : X_j \in X(U_i)\}$.

ALGORITHM RANDOMIZED A

1. Choose $U = U_1, \ldots, U_l$ where U_j has size t and is chosen uniformly in $[k] \setminus A_j$.
2. For each $X \in X(U)$
3. Let
$$O^X = \{\text{argmin}_{v \in P_i} w(v, X_j) : 1 \leq j \leq l, i \in A_j\}.$$
4. Output the core-clique O^X which has minimum weight over all $X \in X(U)$.

We will denote the minimum cost core-clique by O^*.

Lemma 2. *With probability $1 - \delta/2$ over the choice of U there is an $X \in X(U)$ such that $w(O^X) \leq w(O^*) + \epsilon\sigma k^2/2$.*

Proof. For any sequence of samples $U_1, \ldots, U_l$ and $(X_1, \ldots, X_l) \in X(U_1, \ldots, U_l)$, let $S_1, \ldots, S_l$ be defined by
$$S_j = \{\text{argmin}_{v \in P_i} w(v, X_j) : i \in A_j\}.$$
We define a sequence of hybrid core-cliques as follows:
$$O_j = \cup_{i=1}^{j} S_i \cup (\cup_{i=j+1}^{l} V_i \cap O^*).$$
A set $X_j \subseteq W_j$ is *representative for P_i*, where $i \in A_j$ if for all $v \in P_i$,
$$w(v, X_j)/t - w(v, W_j \cap O_{j-1})/|W_j \cap O_{j-1}| \leq \epsilon\sigma/16.$$
A set P_i is homogeneous if for all vertices $v \in P_i$
$$w(v, W_j \cap O_{j-1}) - \min_{u \in P_i} w(u, W_j \cap O_{j-1}) \leq \epsilon\sigma/8;$$
a heterogeneous set is of course a non-homogeneous set. A set $X_j \subseteq W_j$ is *representative* if for all but at most $\epsilon k/8l$ sets P_i where $i \in A_j$, X_j is representative for P_i or P_i is homogeneous. We shall show that with probability $1 - \delta/2l$ over the choice of U_j there is an $X_j \in X(U_j)$ such that
$$w(O_j) \leq w(O_{j-1}) + \epsilon\sigma k^2/l. \tag{1}$$
This immediately implies the lemma. We first show that if $X_j \in X(U_j)$ is representative, then (1) holds. We then show that the probability that there is a representative $X_j \in X(U_j)$ is $1 - \delta/2l$.

Assume that there is an $X_j \subseteq U_j$ which is representative and let S_j be defined as above. Notice the following:

1. The weight of edges between W_j and vertices of sets P_i such that X_j is representative for P_i cannot increase by more than $\frac{\epsilon}{8}\sigma\frac{k}{l}k$.
2. The weight of edges between W_j and vertices of V_j belonging to homogeneous sets P_i cannot increase by more than $\frac{\epsilon}{8}\sigma\frac{k}{l}k$.
3. The weight of edges between W_j and vertices of V_j belonging to heterogeneous sets P_i for which X_j is not representative cannot increase by more than $\frac{\epsilon}{8}\sigma\frac{k}{l}k$, since there are at most $\frac{\epsilon k}{8l}$ such heterogeneous sets P_i.
4. The weight of edges between pairs of vertices of V_j cannot increase by more than $\sigma(\frac{k}{l})^2 = \frac{\epsilon}{8}\sigma\frac{k^2}{l}$.

By Hoeffding's Bound [H63]

$$\Pr_{U_j}[w(v, X_j)/t - w(v, W_j \cap O_{j-1})/|W_j \cap O_{j-1}| > \epsilon\sigma/16] \leq 2^{-\Theta(\epsilon^2 t)} \leq \epsilon\delta/16l.$$

Hence, by Markov's inequality, with probability $1 - \delta/2l$ the sample set U_j is representative.

Algorithm Randomized B

1. Choose $U = U_1, \ldots, U_l$ where U_j has size t and is chosen uniformly in $[k] \setminus A_j$.
2. Uniformly chose a subset $C = \{c_1, \ldots, c_r\}$ of even size $\Theta(\frac{lt\log m + \log(1/\delta)}{\epsilon^2})$ from $[k]$.
3. For each $X \in X(U)$
4. For each $i \in C$, let

$$v_i^X = \text{argmin}_{v \in P_i, 1 \leq j \leq l, i \in A_j} w(v, X_j).$$

5. Output the tuple X which minimize

$$\sum_{i=1}^{r/2} w(v_{2i-1}^X, v_{2i}^X)$$

over all $X \in X(U)$.

The final version of our algorithm does the following. It computes a tuple X using Algorithm Randomized B and then outputs the core-clique $O = \{\text{argmin}_{v \in P_i} w(v, X_j) : 1 \leq j \leq l, i \in A_j\}$. Since

$$2\sum_{i=1}^{r/2} w(v_{2i-1}^X, v_{2i}^X)/r$$

has expected value $w(O^X)/k^2$, it follows that

$$\Pr_C[|2\sum_{i=1}^{r/2} w(v_{2i-1}^X, v_{2i}^X)/r - w(O^X)/k^2| > \epsilon\sigma/4] \leq e^{-\Theta(\epsilon^2 r)} \leq O(\delta m^{-lt}).$$

Since $|X(U)| \leq m^{lt}$, it follows that

$$\Pr_C[\forall X \in X(U), |2\sum_{i=1}^{r/2} w(v_{2i-1}^X, v_{2i}^X)/r - w(O^X)/k^2| < \epsilon\sigma/4] \geq 1 - \delta/2.$$

5 Discussion

This paper has examined a problem from computational biology which arises when one is attempting to perform, for instance, evolutionary studies on molecular sequences. Typically, we are given a large set of homologous sequences partitioned into taxa and we would like to know if there is any evidence of evolutionary relationships between subsets of these taxa. We have also looked at the case where one is given a tree and asked which core-set of vertices from a partite graph best fit to this topology (EXACT-CORE-TREE).

All of these formulations display computational hardness for all reasonable parameterizations and approximation criteria. However, we present a *randomized approximation* algorithm which tests for *min. weight cliqueness* inside of the k-partite graphs, for a given level of confidence and accuracy. All of the algorithms mentioned in this paper have been implemented and tested. We note that our randomized approximation algorithm performs best when the input graph is quite large. We have also tried a number of greedy and randomized greedy heuristics for these problems and we have found that these simple heuristics (like the heuristic for ORTHO-TREE in § 3.3 tend to out-perform our randomized approximation algorithm in practice. There are a number of ways that ideas in the approximation algorithms can be used to derive more advanced heuristics (dominating the simpler ones) and possibly more practical algorithms with proven performance bounds. This is certainly a very challenging line of research that needs further consideration.

References

[A98] Atteson, K. (1998) The performance of neighbor-joining methods of phylogenetic reconstruction. *To be published*, Yale University.

[ABFNPT96] Agarwala, R. et. al. (1996) On the approximability of numerical taxonomy. In: *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, 365–372.

[AGMWL90] Altschul, S. F. et. al. (1990) Basic local alignment search tool. *J. Mol. Biol.*, 215, 403-410.

[BA] Bairoch, A. and Apweiler, R.(1999) The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 1999. *Nuc. Acids Res.*, 27, 49-54.

[B92] Benner, S. A. (1992) Predicting de novo the folded structure of proteins. *Current Opinion in Structural Biology*, 2:402–412.

[B99] Benner, S. A. (1998) Personal communication.

[BDDEHY98] Bork, P. et. al. (1998) Predicting function: from genes to genomes and back. *J. Mol. Biol.* 283, 707–725.

[B71] Buneman, P. (1971) The recovery of trees from measures of dissimilarity. In: *Mathematics in the Archaeological and Historical Sciences*, F. R. Hodson, D. G. Kendall, P. Tauto, eds.: Edinburgh University Press, Edinburgh, 387–395.

[DF99] Downey, R. and Fellows, M. R. (1999) Parameterized Complexity. *Springer Verlag*, New York.

[FHS98] Fellows, M. R. et. al. (1998) On the multiple gene duplication problem. In: *Proceedings of the International Symposium on Algorithms and Computation (ISAAC '98)*, December, Korea.

[FHKS98] Fellows, M. R. et. al. (1998) Analogs & duals of the MAST problem for sequences & trees. *European Symposium on Algorithms (ESA)*.

[F88] Felsenstein, J. (1988) Phylogenies from molecular sequences: inference and reliability. *Annual Revue of Genetics*, 22, 521-565.

[GareyJ79] Garey, M. R. and Johnson, D. S. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, San Francisco.

[GGR98] Goldreich et. al. (1998) Property testing and its connection to learning and approximation. *J. of the ACM*, 45:4, 653–750.

[GCMRM79] Goodman, M. et. al. (1979) Fitting the Gene Lineage into its Species Lineage: A parsimony strategy illustrated by cladograms constructed from globin sequences, *Syst.Zool.*, 28.

[GMS96] Guigó, R. et. al. (1996) Reconstruction of Ancient Molecular Phylogeny. *Molec. Phylogenet. and Evol.*, 6(2), pp. 189–213, 1996.

[HL99b] Hallett, M. T. and Lagergren. J. (1999) ¿From gene trees to species tree for a bounded number of orthologues. Manuscript in preparation. *To be published.*

[H63] W. Hoeffding. (1963) Probability inequalities for sums of bounded random variables. *Amer. Statist. Assoc. J.*, 58, 13–30.

[KTG98] Koonin, E. V. et. al. (1998) Beyond complete genomes: from sequence to structure and function. *Curr Opin Struct Biol*, 8(3), 355-63.

[MLZ98] Ma, B. et. al. (1998) On Reconstructing Species Trees from Gene Trees in Term of Duplications and Losses. *Recomb 98*.

[MMS95] Mirkin, B. et. al. (1995) A biologically consistent model for comparing molecular phylogenies. *Journal of computational biology*, 2(4), 493–507.

[P98] Page, R. (1998) GeneTree: comparing gene and species phylogenies using reconciled trees. *Bioinformatics*, 14(9), 819–820.

[PC97] Page, R. and M. Charleston, M. (1997) ¿From Gene to organismal phylogeny: reconciled trees and the gene tree/species tree problem. *Molec. Phyl. and Evol. 7*, 231–240.

[PL88] Pearson, W. R. and Lipman, D. J. (1988) Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci.*, 85:2444-2448.

[RS96] Rubinfeld, R. and Sudan, M. (1996) Robust characterization of polynomials with applications to program testing. *SIAM J. Comput.* 25, 2, 252-271.

[SN87] Saitou, N. and Nei, M. (1987) The neighbour-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, 4, 406–425.

[SM98] Slonimski et. al. (1998) The first law of genomics. *Abstract "Microbial Genomes II"*, Hilton Head, January.

[S99] Stoesser, G. et. al. (1999) The EMBL Nucleotide Sequence Database. *Nuc. Acids Res.*, 27(1), 18-24.

[TKL97] Tatusov, R. L. et. al. (1997) A genomic perspective on protein families. Science, 278(5338), 631-7.

[YEVB98] Yuan, Y. P. et. al. (1998) Towards detection of orthologues in sequence databases. *Bioinformatics*, 14(3), 285–289.