Quantifying Hierarchical Conflicts in Homology Statements

Krister M. Swenson, Afif Elghraoui, Faramarz Valafar, Siavash Mirarab, Mathias Weller

Quantifying Hierarchical Conflicts in Homology Statements

Krister M. Swenson[1,2][0000−0001−8690−1261], Afif Elghraoui[3][0000−0002−6489−9444], Faramarz Valafar[5][0000−0002−3648−9384], Siavash Mirarab[4][0000−0001−5410−1518], and Mathias Weller[1,6][0000−0002−9653−3690]

[1] Centre National de la Recherche Scientifique (CNRS)
[2] LIRMM, University of Montpellier, France
swenson@lirmm.fr
[3] Dept. of Electrical and Computer Engineering, San Diego State University, San Diego, CA
aelghraoui@sdsu.edu
[4] Dept. of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA
smirarab@ucsd.edu
[5] School of Public Health, San Diego State University, San Diego, CA
faramarz@sdsu.edu
[6] LIGM, Université Gustave Eiffel, Paris, FRANCE
mathias.weller@cnrs.fr

Abstract. A fundamental step in any comparative whole genome analysis is the annotation of homology relationships between segments of the genomes. Traditionally, this annotation has been based on coding segments, where orthologous genes are inferred and then syntenic blocks are computed by agglomerating sets of homologous genes into homologous regions. More recently, whole genomes, including intergenic regions, are being aligned *de novo* as whole genome alignments (WGA). In this article we develop a test to measure to what extent sets of homology relationships given by two different software are hierarchically related to one another, where matched segments from one software may contain matched segments from the other and *vice versa*. Such a test should be used as a sanity check for an agglomerative syntenic block software, and provides a mapping between the blocks that can be used for further downstream analyses. We show that, in practice, it is rare that two collections of homology relationships are perfectly hierarchically related. Therefore we present an optimization problem to measure how far they are from being so. We show that this problem, which is a generalization of the assignment problem, is NP-Hard and give a heuristic solution and implementation. We apply our distance measure to data from the Alignathon competition, as well as to *Mycobacterium tuberculosis*, showing that many factors affect how hierarchically related two collections are, including sensitivities to guide trees and the use or omission of an outgroup. These findings inform practitioners on the pitfalls of homology relationship inference, and can inform development of more robust inference tools.

Keywords: Homology · Syntenic Block · T-Star Packing · Assignment Problem

1 Introduction

The increasing ease of whole genome sequencing and assembly has opened a new era of comparative genomics. With the data available today, not only can the phylogenetic histories of all the genes between a set of genomes be analyzed, but also the interaction between these genes, linking gene regulation and function to the positions of groups of genes. These analyses require a reliable grouping of homologous genomic segments from the multiple genomes in question.

Thus, the inference of sets of homologous genomic segments is of fundamental importance. Such a *homology statement* comes in the form of a set of genomic segments that contains at least one, but potentially multiple, segments from several genomes. Each pair of segments from the set shares common ancestry over some proportion of their intervals, which varies depending on the scale and level of precision required by the application.

The most basic segment on which statements are made has traditionally been the gene, detected through either manual or automatic means. The number of tools designed to infer homology relationships between annotated genes has grown, provoking the formation of the Quest for Orthologs (QfO) consortium dedicated to the evaluation and comparison of these tools [12].

General genomic intervals, that can contain both coding and noncoding positions, have also been used as homology statements. In this case, researchers have considered bidirectional best hits as evidence for orthology [22,31]. More recently, "whole genome alignment" methods partition entire genomes into blocks that can be aligned into multiple sequence alignments (MSAs), *de novo*, with no special input from the user. The Alignathon collaborative competition was developed to evaluate and compare these methods [9].

Study of the large scale changes between genomes has inspired a more vague notion of homology between genome segments. Even before the discovery of the double helix, groups were studying homology of large segments of genomes from the salivary glands of drosophila [29]. More precise lengths of roughly *conserved chromosomal segments* began to be studied using linkage maps [23]. In the postgenomic era, basic homology segments are agglomerated into *syntenic blocks*, possibly separated by micro-rearrangements. GRIMM-synteny was developed for the study of large scale chromosomal changes, in response to the whole genome sequencing efforts in human and mouse [27]. Since then, many syntenic block inference tools have been introduced but, despite twenty years of development, a unified definition of syntenic block has yet to be found. Indeed, most tools rely on operational definitions rather than biological or mathematical ones [11,30].

There are several tests used for comparing and evaluating homology statements. For orthology statements between coding sequences, the QfO project has established tests that: 1) compare trees inferred from orthologous families to agreed-upon species trees, 2) compare the subsets of orthologs from curated gene

families, and 3) use consistency in gene ontology annotation [1]. For statements between syntenic blocks, comparisons of inclusion and exclusion of segments between methods have been done, and blocks from a single method have been compared to known gene clusters [19]. Other sources of ground truth, such as RNA-seq data, have been used to confirm co-regulation between genes occurring in a proposed block [33].

For general statements (on both coding and noncoding DNA), the Alignathon competition used three different measures [9]. If homology statements are given as a set of (potentially gapped) equal-length segments from several genomes, then each homologous pair of positions between two genomes as given by one method, can be queried in another method. The number of such shared positions is a measure of similarity and, when one of the methods is taken as ground truth, the number of shared positions can be used to measure precision and recall. The `mafComparator` tool estimates these values by sampling positions [9]. For sets of aligned homology statements (*i.e.* MSAs), *probabilistic sampling-based alignment reliability* (PSAR) was used to assess each aligned column [16]. PSAR fixes all rows of the alignment but one, and samples from the many ways to align that row within the fixed alignment. After this is repeated for each row, an alignment reliability score for each pair of positions in a column can be assigned. When aligned homology statements are augmented with a phylogeny, another statistical test called `StatSigMA` can be used [28]. For each edge of the phylogeny, the rows of the alignment are split into two alignments. The two alignments are then tested for exhibiting "unrelated behaviour" using Karlin-Altschul log likelihood scores. If the test for all branches passes, then the homology statement is validated.

For homology statements that come in the form of syntenic blocks, Ghiurcuta and Moret outline some necessary conditions for a valid agglomeration of homologous units into such blocks [11].

There exists very few methods that compare homology statements in the form of sets of genomic segments, unmarried to connotations of orthologous genes, and independent of multiple sequence alignments. To our knowledge, the Jaccard distance (e.g. as computed by `mafComparator`) applied to pairwise homology statements, is the only known comparison that falls into this category.

In this article we introduce a simple definition of *homology block* (Section 2.1) and formally characterize the conditions under which a set of homology blocks are valid (Section 2.2). We show what it means for collections of blocks to be hierarchically related and use this to develop a method for measuring disagreement between two different collections of blocks. In Section 2.3 we show a necessary condition for two collections of blocks to be in a hierarchical relationship (in the form of Lemma 1), based on a graph representing the overlap between the sets. For different parts of the genomes in question, our test allows for the collections of blocks to be hierarchically related in both ways; in some parts of the genomes the first set could be more general than the second, while in other parts of the genomes the opposite can be true. We introduce an optimization problem, called Minimum Deletion into Disjoint Stars (MDDS), which gives a lower bound

on the number of positions that must be ignored so that the two collections of blocks could be related through a hierarchical relationship. Not only does a solution to MDDS give a measure as to the degree of hierarchical dissonance between two collections, but it serves as an unambiguous mapping between the blocks of the two. This mapping could be used for further downstream comparisons, an illustration of which is shown in Appendix C.

We show that the MDDS problem is NP-Hard, before presenting a polynomial time heuristic based on an exact algorithm for solving MDDS on a tree. In Section 4 we define the *homology discordance ratio* and use this measure as a distance between block collections built on Alignathon data and on a set of 94 *Mycobacterium tuberculosis* isolates. On the tuberculosis strains we study the relationship between blocks built using an outgroup or no outgroup, using annotations or no annotations, using `maf2synteny` to agglomerate or not, as well as study the effect of the guide tree on `Cactus` MSA blocks. On the simulated data from the Alignathon project, we highlight differences between our method and the Jaccard distance (as computed with `mafComparator`).

For the entirety of the article we focus on the general case of homology statements, although most of the discussion also applies to the restricted case of orthology statements.

2 Methodological Foundations

2.1 Overlapping Homology Statements and the Block Graph

We use $g_i[k..\ell]$ to denote a *segment* between positions k and ℓ of genome g_i and we let T denote the universe of all segments over all possible genomes and position-pairs. Define the overlap $op(s_1, s_2)$ between two segments s_1 and s_2 as the number of positions where they overlap in the same genome. For example $op(g_1[1..5], g_1[3..9]) = 3$ but $op(g_1[1..5], g_2[3..9]) = 0$.

Definition 1 (homology statement block). *A* homology statement block *(called a* block *for short) B is a set of segments $B \subset T$ such that all pairs of segments in B have zero overlap.*

The right panel of Fig. 1 depicts two collections of homology statement blocks $\mathcal{A} = \{A1, A2, A3\}$ and $\mathcal{B} = \{B1, B2, B3\}$. The blocks of $\mathcal{A}$ are $A1 = \{g_1[1..12], g_2[13..24]\}$, $A2 = \{g_1[14..28], g_2[26..41]\}$, $A3 = \{g_1[53..66], g_2[101..113]\}$, while the blocks of $\mathcal{B}$ are $B1 = \{g_1[1..28], g_2[13..41]\}$, $B2 = \{g_1[46..68], g_2[94..115]\}$, and $B3 = \{g_1[34..39], g_2[42..46], g_2[116..121]\}$.

Before discussing the semantic interpretation of a homology statement block, we first introduce a graph that represents the overlap between blocks. The overlap $op(B1, B2) = \sum_{s_1,s_2 \in B1 \times B2} op(s_1, s_2)$ between blocks $B1$ and $B2$ is the total overlap between all pairs of segments in the two. A collection of blocks $\mathcal{B} = \{B1, B2, \ldots\}$ is considered to be *clean* if the overlap between any pairs of blocks in $\mathcal{B}$ is zero. Both collections depicted in Fig. 1 are clean.

For two collections of blocks $\mathcal{A}$ and $\mathcal{B}$, we build a bipartite *block graph* $BG(\mathcal{A}, \mathcal{B})$ where there is an edge between A and B for any $A \in \mathcal{A}$ and $B \in \mathcal{B}$ if

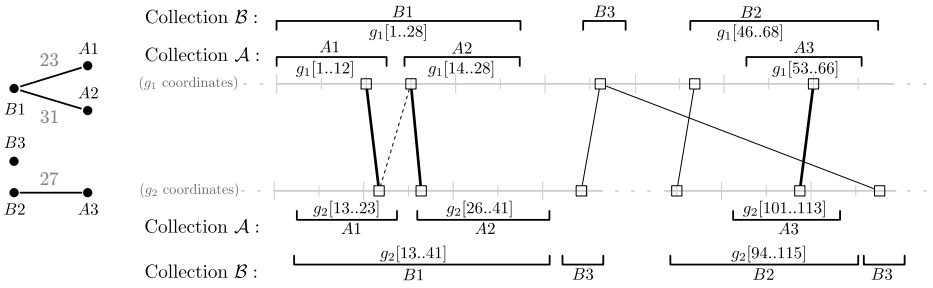

Fig. 1: **To the right**, the collections of blocks $\mathcal{A} = \{A1, A2, A3\}$ and $\mathcal{B} = \{B1, B2, B3\}$ appearing in genomes g_1 and g_2, along with the graph $BG(\mathcal{A}, \mathcal{B})$. Segments are depicted with brackets and lined up according to their positions on the chromosomes. They are labeled by their tuple (when space permits) and the block to which they belong. The configuration of positive and negative witness pairs shows that $\mathcal{B}$ generalizes $\mathcal{A}$. Some of the genome positions are highlighted with boxes, and two such positions are connected by a solid line if they appear as a positive witness in $\mathcal{B}$, and that line is bold if they are also in $\mathcal{A}$. The dashed line represents one of the (many) negative homology witnesses between $A1$ and $A2$ that are negative witness pairs for $\mathcal{A}$ but not for $\mathcal{B}$. **To the left**, the graph $BG(\mathcal{A}, \mathcal{B})$ appears with edges labeled by overlap length in gray. All of the connected components are stars.

and only if blocks A and B overlap (*i.e.* $op(A, B) > 0$). Thus, the block graph $BG(\mathcal{A}, \mathcal{A})$ for a clean collection $\mathcal{A}$ is a perfect matching. $E(G)$ is the set of edges of the graph G. We associate to each edge $AB \in E(G)$ a weight function $\omega : E(G) \to \mathbb{N}$ such that $\omega(AB) = op(A, B)$. The left side of Fig. 1 shows the block graph for the collections to the right.

2.2 Homology Witnesses and Block Hierarchies

A homology block can be interpreted as a *positive* and *negative* statement of homology (*i.e.* statements about common ancestry). On the positive side, the block $\{g_1[1..5], g_2[11..16]\}$ says that positions 1 through 5 in genome g_1 are somehow homologous to positions 11 through 16 in genome g_2 (in this case the segments are not the same length, so we assume that each position from the segment $g_1[1..5]$ is homologous to either a position in $g_2[11..16]$ or to none other in g_2). On the negative side, the block could be interpreted as saying that *no* position in $g_1[1..5]$ is homologous to any other position in g_1 or any other position in g_2, outside of $g_2[11..16]$.

In this section we suppose that we know the truth about the ancestral relationships between the base-pair positions of the genomes in question. With this supposed knowledge, we can categorize pairs of *homology witness* positions as positive or negative, depending on their evolutionary relationship. Using these relationships, we define properties that a valid collection of homology blocks must

respect. These definitions are extended to encompass hierarchical relationships between collections of blocks.

Consider any pair of positions $g_i[x]$ and $g_j[y]$ from genomes g_i and g_j. This pair is called a *positive homology witness* if the two positions descend from the same ancestral position, otherwise the pair is called a *negative homology witness* (positive homology witnesses represent pairs of positions that are typically called "homologous" positions). Note that the true relationship between positions is unknown, yet it imposes constraints on what we consider a valid collection of blocks according to the following definition.

Consider any position-pair $(g_i[x], g_j[y])$ such that $g_i[x]$ is contained in a segment from a block B in a collection $\mathcal{B}$. If $(g_i[x], g_j[y])$ is a positive homology witness, then either

1. $g_j[y]$ appears in B, and we say that the pair is a *positive witness in* $\mathcal{B}$, or
2. $g_j[y]$ appears in no block of $\mathcal{B}$.

By this definition no position-pair $(g_i[x], g_j[y])$ with $g_i[x]$ and $g_j[y]$ in different blocks of $\mathcal{B}$, can be a positive homology witness and, since all position-pairs are either positive or negative homology witnesses, $(g_i[x], g_j[y])$ must be a negative homology witness. Any position-pair $(g_i[x], g_j[y])$, where $g_i[x]$ and $g_j[y]$ are in different blocks of $\mathcal{B}$ or in no block of $\mathcal{B}$, is called a *negative witness for* $\mathcal{B}$.

Note that, for a clean collection $\mathcal{B}$, no position-pair can be both a positive and negative witness in/for $\mathcal{B}$. There may also be position-pairs that are neither positive nor negative witnesses in/for $\mathcal{B}$, such as those pairs that have one position contained in a block of $\mathcal{B}$ and the other outside all blocks of $\mathcal{B}$. Finally, note that not all position-pairs appearing between segments in a homology block need necessarily be positive homology witnesses.

Positive witness pairs limit what can exist in two different blocks; a block containing one position of a positive homology witness imposes the constraint that the other position must either be in the same block, or in no block. On the other hand, we will see in the following that negative homology witnesses existing between two different blocks in a collection enforce constraints on the hierarchical relationships that this collection can have with another block collection.

Consider the collections of blocks in Fig. 1 and note that whenever a positive homology witness is a positive witness in $\mathcal{A}$, it must also be a positive witness in $\mathcal{B}$, whereas not all positive witnesses in $\mathcal{B}$ exist in $\mathcal{A}$. Conversely, every negative witness for $\mathcal{B}$ is also a negative witness for $\mathcal{A}$. In this sense, the blocks of $\mathcal{B}$ are "more general" than the blocks of $\mathcal{A}$. This motivates the following definition, for which we focus on subcollections of blocks $\mathcal{A}' \subseteq \mathcal{A}$ and $\mathcal{B}' \subseteq \mathcal{B}$.

Definition 2 (generalization). *A clean (sub)collection of blocks* $\mathcal{B}'$ generalizes *a clean (sub)collection* $\mathcal{A}'$ *if and only if every positive witness in* $\mathcal{A}'$ *is also a positive witness in* $\mathcal{B}'$ *and every negative witness for* $\mathcal{B}'$ *is also a negative witness for* $\mathcal{A}'$.

Note that any clean collection generalizes itself.

While some subcollections of $\mathcal{B}$ may generalize subcollections of $\mathcal{A}$, other subcollections of $\mathcal{A}$ may generalize subcollections of $\mathcal{B}$. Partition them $\mathcal{A} =$

$\mathcal{A}_1 \cup \mathcal{A}_2 \cup \cdots \cup \mathcal{A}_k$ and $\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \cdots \cup \mathcal{B}_k$ according to the connected components of $BG(\mathcal{A}, \mathcal{B})$ (*e.g.* $\mathcal{A}_1 \cup \mathcal{B}_1$ is the set of vertices in the first connected component).

Definition 3 (hierarchical). *We say that $\mathcal{A}$ and $\mathcal{B}$ have a* hierarchical relationship *if and only if $\mathcal{A}_i$ generalizes $\mathcal{B}_i$, or $\mathcal{B}_i$ generalizes $\mathcal{A}_i$, for $1 \leq i \leq k$.*

The existence of hierarchical relationships between collections of blocks are interesting to us for at least two reasons. Consider two block inference methods, MethodA and MethodB, producing different clean collections of blocks $\mathcal{A}$ and $\mathcal{B}$ respectively. If MethodB is meant to agglomerate blocks from MethodA, then we would expect $\mathcal{B}$ to generalize $\mathcal{A}$. This is useful for the verification of agglomeration methods, and as a sanity check for the practitioner. In this case, if MethodB also trims spurious blocks or segments from MethodA, $\mathcal{B}$ may not generalize $\mathcal{A}$, but $\mathcal{A}$ and $\mathcal{B}$ would still be hierarchically related. Another reason for interest in the hierarchical relationship may be that, if $\mathcal{B}$ generalizes $\mathcal{A}$, then we can define a mapping from each block $A \in \mathcal{A}$ to a block $B \in \mathcal{B}$. This mapping can be used for further comparisons between the collections. The refinement of orthology assignments, as illustrated in Appendix C is an example of one such comparison.

2.3 Relating Block Hierarchy to Stars in the Block Graph

While simple hierarchical relationships are easy to detect, real-world data are not so well behaved, and require a formalism to measure the extent to which a relationship is hierarchical. The types of connected components in the block graph give us insight into collections that cannot have a hierarchical relationship.

Lemma 1. *Let $\mathcal{A}$ and $\mathcal{B}$ be clean collections of blocks such that $\mathcal{B}$ generalizes $\mathcal{A}$ and $BG(\mathcal{A}, \mathcal{B})$ is connected. Then, all vertices of $\mathcal{A}$ have degree one in $BG(\mathcal{A}, \mathcal{B})$, that is, $BG(\mathcal{A}, \mathcal{B})$ is a star with center in $\mathcal{B}$.*

Proof. Let A be a block in $\mathcal{A}$, and assume that it has at least two distinct neighbors $B1, B2 \in \mathcal{B}$ in $BG(\mathcal{A}, \mathcal{B})$, that is, both $B1$ and $B2$ overlap A. Thus, there are positions $g_i[x]$ and $g_j[y]$ appearing in A such that $g_i[x]$ appears in $B1$ and $g_j[y]$ appears in $B2$. Since $\mathcal{B}$ is clean, we also know that these positions are distinct. Since $(g_i[x], g_j[y])$ appears in different blocks in $\mathcal{B}$, we know that it is a negative homology witness for $\mathcal{B}$. However, since $g_i[x]$ and $g_j[y]$ appear in the same block in $\mathcal{A}$, the pair is not a negative witness for $\mathcal{A}$. Thus, $(g_i[x], g_j[y])$ is a negative witness for $\mathcal{B}$ but not for $\mathcal{A}$, contradicting the fact that $\mathcal{B}$ generalizes $\mathcal{A}$. □

We say that a graph is *hierarchical* if it is a collection of vertex-disjoint stars, that is, if no component has two vertices of degree greater than one. It is easy to check if a graph meets this criterion. Note that the condition of a graph $BG(\mathcal{A}, \mathcal{B})$ being hierarchical is necessary for $\mathcal{A}$ and $\mathcal{B}$ to have a hierarchical relationship, but it is not sufficient. Also note that, Lemma 1 outlines a property on each individual connected component, allowing some parts of $\mathcal{A}$ to generalize parts

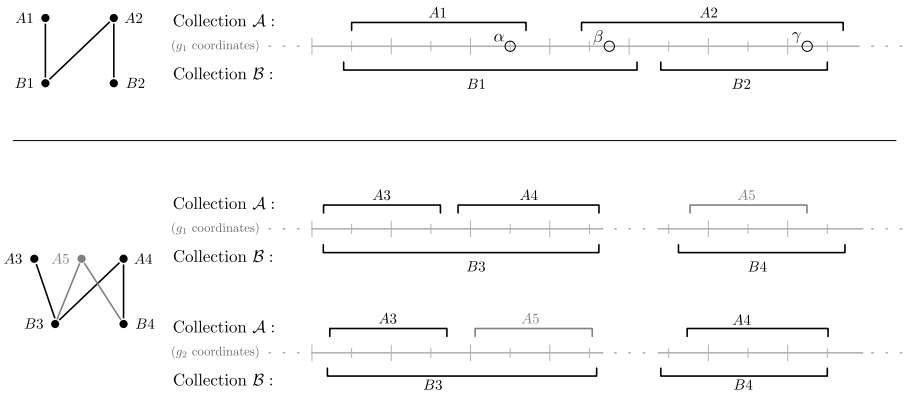

Fig. 2: Two subsets of blocks from collections $\mathcal{A}$ and $\mathcal{B}$ that are not hierarchically related. The **top panel** depicts a subset of blocks for part of the genome g_1, but not the other genomes. Positions α and β form a negative homology witness for $\mathcal{A}$, but not for $\mathcal{B}$, while β and γ form a negative homology witness for $\mathcal{B}$, but not for $\mathcal{A}$. This contradicts properties of a hierarchy and, therefore, yields the non-star topology to the left. The **bottom panel** depicts a subset of blocks for part of the genomes g_1 and g_2. These segments contradict properties of a hierarchy in a different way, and yield the non-star topology to the left. This kind of scenario would arise when orthologs are matched in one way from MethodA, and in another way for MethodB. Note that even if the block $A5$ was not in collection $\mathcal{A}$, the contradiction still holds and the connected component is not a star.

of $\mathcal{B}$, while allowing other parts of $\mathcal{B}$ to generalize parts of $\mathcal{A}$. Thus, a natural corollary to Lemma 1 is that if $\mathcal{A}$ and $\mathcal{B}$ are hierarchically related, then $BG(\mathcal{A}, \mathcal{B})$ is hierarchical.

For a graph that is *not* a collection of stars, one may want to measure to what degree it deviates from being so. Lemma 1 inspires the search for *star packings* on $G = BG(\mathcal{A}, \mathcal{B})$.

Minimum Deletion into Disjoint Stars (MDDS)
Input: a bipartite graph G with weight function $\omega : E(G) \to \mathbb{N}$
Output: $E' \subseteq E(G)$ such that the subgraph of G formed by the edge set $(E(B) \setminus E')$ is a collection of vertex-disjoint stars
Measure: $\sum_{e \in E'} \omega(e)$

A solution to MDDS gives a lower bound on the number of overlapping positions that must be ignored so that $\mathcal{A}$ and $\mathcal{B}$ can be hierarchically related. For example, Fig. 2 shows two connected components that are not stars. Consider the graph from the upper panel and assume that in the non-depicted genomes (*i.e.* g_i for $i > 1$) there is no overlap of the segments of $A1$ with those of $B2$, or of

segments of $A2$ with those of $B1$. Then, the solution to MDDS on this component would result from the removal of the edge between $B1$ and $A2$, since this edge has the minimum-size overlap. In Section 4.1 we highlight the differences between our MDDS method, and the Jaccard similarity index used in the Alignathon.

3 Algorithms

In this section we show that the Minimum Deletion into Disjoint Stars problem is NP-Hard, and then present a practical heuristic based on solving MDDS optimally on a tree. For simplicity and without loss of generality, we will assume that all block graphs are connected.

Other generalizations of the assignment problem, similar to MDDS, have been studied for decades, the closest of which has most recently been called the T-star packing problem [4]. This problem asks for a star packing where the size of the star is limited by an input parameter T. When the measure is the number (or weight) of edges, Hell and Kirkpatrick show that the T-star packing is NP-Hard by reduction to the version of the problem that asks for a decomposition of a given graph into subgraphs isomorphic to the star with T edges [13]. Since the only difference between MDDS and the edge-weighted T-star packing problem is the parameter T, it is tempting to adapt the same series of reductions to MDDS by setting T to the maximum degree over all vertices in the graph. This approach is not clearly feasible, however, since the reduction from 3-Dimensional Matching to the decomposition version of the problem creates vertices of degree higher than T [17].

Babenko and Gusakov give a $\frac{9}{4}\frac{T}{T+1}$ approximation algorithm for the T-star packing problem based on a reduction to the max-network flow problem [4]. We could use this elaborate approximation algorithm by fixing T to the maximum degree of the input graph, but we choose instead to implement the much simpler heuristic presented in Section 3.2.

3.1 NP-Hardness of MDDS

We will show that the decision version of MDDS is NP-hard by reducing the well-known 3-SAT problem to it. Our construction uses similar techniques as the NP-hardness proof of the Transitivity Edge Deletion problem [32].

Construction 1 (see Fig. 3) *Let φ be an instance of* 3-SAT *with variables $X := \{x_1, x_2, \ldots, x_n\}$ and clauses $\mathcal{C} := \{C_1, C_2, \ldots, C_m\}$ such that each clause contains exactly three literals. For each variable x_i, let n_i denote the number of clauses that contain x_i or $\neg x_i$ and let $\gamma_i^0, \gamma_i^1, \ldots, \gamma_i^{n_i-1}$ be any sequence of these clauses. We construct an edge-weighted graph (G, ω) as follows:*

1. *For each variable x_i create a cycle Q_i containing $6n_i$ vertices $v_i^0, v_i^1, \ldots, v_i^{6n_i-1}$ and give all edges weight m.*
2. *For each clause $C_k \in \mathcal{C}$, create a single vertex u_k.*

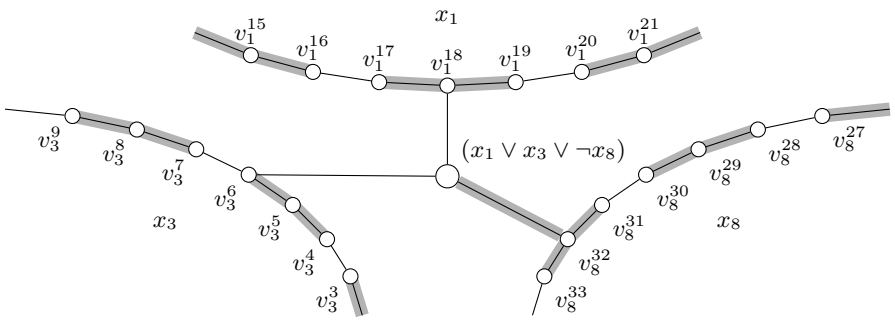

Fig. 3: Example of Construction 1. The clause $C := (x_1 \vee x_3 \vee \neg x_8)$ corresponding to the center vertex is equal to $\gamma_1^3 = \gamma_3^1 = \gamma_8^5$, that is, it is the 4^{th} clause containing x_1, the 2^{nd} clause containing x_3 and the 6^{th} clause containing x_8. A truth-assignment setting x_1 to TRUE and x_3 and x_8 to FALSE corresponds to the star cover indicated by gray highlights. Note that taking the edge between v_1^{18} and C instead of the edge between v_8^{32} and C corresponds to satisfying the clause C by x_1 instead of $\neg x_8$.

3. *For each i, j let ℓ be such that $\gamma_i^j = C_\ell$ and, if x_i occurs non-negated in C_ℓ, then add the edge $\{v_i^{6j}, u_\ell\}$ with weight 1, otherwise add the edge $\{v_i^{6j+2}, u_\ell\}$ with weight 1.*

Note that the image of ω is $\{1, m\}$, the total weight of all edges is $18m^2 + 3m$, and G is bipartite, since any edge from part 3 of the construction, connecting a (variable) cycle to a (clause) vertex u_ℓ, connects to an even numbered vertex in the cycle.

Besides NP-hardness, our reduction implies exponential lower bounds assuming widely believed complexity-theoretic hypotheses. The "Exponential-Time Hypothesis" (combined with results by Impagliazzo et al. [15]) roughly states that 3SAT on formulas with m clauses cannot be decided in $2^{o(m)}$ time. This lower bound transfers since the constructed graph G has only $21m$ edges.

Theorem 1. Minimum Deletion into Disjoint Stars *is NP-hard and cannot be solved in $2^{o(|E(G)|)}$ time on graphs G, even if G is restricted to maximum degree three, assuming the Exponential-Time Hypothesis.*

3.2 A Heuristic for MDDS

In light of the hardness result presented in Section 3.1, we devised a heuristic that first computes a maximum-weight spanning tree T on each connected component of BG. It then transforms each T into a star packing by computing MDDS on T.

We present a dynamic programming algorithm solving MDDS on a tree T. To this end, we root T at an arbitrary vertex, and compute a dynamic programming table for vertices in a post-order traversal. Consider a set S of edges that, after

removal from T, yields a collection of disjoint stars. We denote the result of this removal as $T - S = (V(T), E(T) \setminus S)$. Each vertex x has one of three states relative to S:

1. x is the center of a star in $T - S$ (covered by $D_*(x)$ in the DP),
2. x has degree one in $T - S$ and the edge between x and its parent is in $T - S$ (covered by $D_+(x)$ in the DP), and
3. x has degree one in $T - S$ and the edge between x and its parent is not in $T - S$ (covered by $D_-(x)$ in the DP).

Then, $D_*(x)$, $D_+(x)$, and $D_-(x)$ contain the weight of an optimal solution S_x for the subtree rooted at x, for each of the three cases respectively. If x is a leaf of T, then set $D_*(x) := D_-(x) := D_+(x) := 0$. Otherwise, let $v_1, v_2, \ldots, v_m$ denote the children of x in T. We visit the children in this order and accumulate each partial subsolution, starting with $D^0_*(x) := D^0_+(x) := D^0_-(x) := 0$ and proceeding as follows for each $1 \leq i \leq m$:

$$D^i_*(x) := D^{i-1}_*(x) + \min\left(D_+(v_i),\ \omega(xv_i) + \min(D_*(v_i), D_-(v_i))\right)$$

That is, if x is the center of a star, then the edge xv_i must be in S if either v_i is the center of a star or the edge between v_i and its parent x is not in $T - S$.

$$D^i_+(x) := D^{i-1}_+(x) + \omega(xv_i) + \min(D_*(v_i), D_-(v_i))$$

That is, if x is a leaf of a star centered at the parent of x, then the edge xv_i must be in S.

$$D^i_-(x) := \min\begin{pmatrix} D^{i-1}_-(x) + \omega(xv_i) + \min(D_*(v_i), D_-(v_i)), \\ D^{i-1}_+(x) + D_*(v_i) \end{pmatrix}$$

The case of $D^i_-(x)$ is a bit more subtle. Since x is not the center of a star, all but at most one edge between x and its children are in S, so if xv_i is not in S then $D^{i-1}_+(x)$ forces all xv_j to be in S, for $1 \leq j < i$. Finally, the subsolutions rooted at x are, then, given by:

$$D_+(x) := D^m_+(x) \qquad D_-(x) := D^m_-(x) \qquad D_*(x) := D^m_*(x)$$

4 Quantifying Hierarchical Conficts

We applied our MDDS heuristic to homology statements on a set of prokaryotes, and on a set of eukaryotes. The solution to MDDS provides an estimate of the minimum number of positions that must be ignored so that the necessary conditions for a hierarchy, highlighted by Lemma 1, are achieved. Before applying the heuristic of Section 3.2 we cleaned the syntenic blocks according to Appendix B, and preprocessed the graphs for segmental duplications according to Appendix C.

4.1 Discordance Ratio and Distinction from Jaccard Index

Define *coverage($\mathcal{B}$)* of a collection of blocks $\mathcal{B}$ as the total number of positions covered by all segments in blocks of $\mathcal{B}$. We report the *hierarchical discordance ratio* between collections $\mathcal{A}$ and $\mathcal{B}$ as $d(\mathcal{A},\mathcal{B}) = w/(coverage(\mathcal{A})+coverage(\mathcal{B}))$, where w is the weight of the MDDS on $BG(\mathcal{A},\mathcal{B})$. A discordance ratio of 0.1 means that we have to ignore at least 10% of the total coverage of the blocks (in both methods) in order to have a hierarchical relationship between them.

Alignathon used `mafComparator` to compute the straightforward Jaccard similarity index between collections of blocks. In this case, the elements of the sets in question are the pairwise alignments of positions implied by the blocks. So if a pair of positions are aligned in one class of blocks but not the other, this will contribute one to the denominator.

Consider collections $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{A}$ only contains blocks with segments from $\{g_i[200x+1..200x+100] \mid 0 \leq x < \lfloor \frac{\ell(g_i)}{200} \rfloor\}$, and $\mathcal{B}$ only contains blocks with segments from $\{g_i[200x+101..200x+200] \mid 0 \leq x < \lfloor \frac{\ell(g_i)}{200} \rfloor\}$, for all genomes g_i with length $\ell(g_i)$. In other words, the collections can only have blocks of length 100 that do not overlap with each other. In this case the Jaccard similarity measure will be zero no matter the length of the genomes, indicating the most severe dissimilarity, whereas the two collections are hierarchically related, showing no conflicts, and the block graph is composed only of degree zero vertices. In that sense, our comparison method is tolerant to collections that conservatively make no assertion about a region.

We consider the two measures complementary in that they capture different qualities of the overlap properties of block collections. We see in Section 4.3 instances from the Alignathon data where the Jaccard similarity is low, yet the two collections are hierarchically related, and *vice versa*.

4.2 *Mycobacterium tuberculosis* clinical isolates

For the prokaryotes, we used a set of 94 *Mycobacterium tuberculosis* strains [21,5] with homology statements given by the methods listed in Table 1. These sets of blocks are those produced in [10], where the methods were compared to assess their impact on inferring rearrangement phylogenies. Note that all `Cactus` blocks used in this subsection had segments with fewer than 50 positions filtered out.

The collections of blocks for four of the methods, along with their `maf2synteny` counterparts are compared in Fig. 4. As expected, each collection of blocks had a very low discordance ratio with its counterpart agglomerated by `maf2synteny`. Further, the agglomerated blocks always have lower discordance to all the other methods, when compared to their unagglomerated counterparts. Of the unagglomerated methods, `SibeliaZ` is the least discordant.

There were a couple of surprises. The first is that the most discordant pairs are between the gene-based annotation method and the *de novo* inference methods `Cactus` and `SibeliaZ`. Contrary to the other methods, the agglomerated annotation blocks show a small improvement against `Cactus`, and a surprising degradation (going up from 4% to 5%) in the discordance ratio for `SibeliaZ`.

Table 1: Homology statement determination methods applied to the *M. tuberculosis* genomes

Method	Description
`Cactus(SNP)`	`Cactus` [3] alignment guided by a ML tree based on concatenated substitutions with respect to reference strain H37Rv (NCBI accession NC_000962.3)
`Cactus(SibeliaZ)`	`Cactus` alignment guided by a MLWD[14] adjacency tree computed from `SibeliaZ+M2S` synteny blocks.
`Cactus(Mash)`	`Cactus` alignment guided by a B(I)ONJ tree based on the genomes' Mash[25] distance matrix.
`SibeliaZ`	Locally collinear blocks produced in the first step of the `SibeliaZ` pipeline[20].
`Annotation`	Simultaneous annotation and orthology assignment by 95% amino acid sequence identity and 95% alignment coverage.

Modifiers	Description
`+out`	Synteny blocks computed while including the outgroup strain *M. canettii* (NCBI accession NC_019951.1)
`+M2S`	Agglomerated with `maf2synteny` [18]

This implies that either 1) many blocks from the `Cactus` method bridge between coding regions, or 2) many duplicate regions are assigned in discordant ways. The second surprise is that the unagglomerated `Cactus` methods, with different guide trees, are more discordant from each other than they are with `SibeliaZ`. It has been reported that `Cactus`'s sensitivity to guide trees also has implications on the downstream phylogenetic analyses [10].

In Fig. 5, the checkered pattern shows that the inclusion of an outgroup affects `Cactus` blocks more than the choice of a guide tree. The inclusion of the outgroup strain also decreases the discordance between the `Cactus` blocks on different guide trees. For example, `Cactus(Mash)` has discordance ratios of 0.044 and 0.061 against `Cactus(SNP)` and `Cactus(SibeliaZ)`, but for `Cactus(Mash)+out` these values are 0.022 and 0.026. Table 2 shows the discordance between a method and its version with the outgroup. `Cactus` is most highly affected by the inclusion of the outgroup. While `SibeliaZ` is somewhat affected, Annotation is barely affected. Agglomerating the blocks with `maf2synteny` diminishes the discordance in all cases but Annotation.

4.3 Alignathon

The Alignathon competition was created to compare "whole genome alignment" methods [9]. Authors of WGA software were invited to submit the collections of blocks computed by their program, which were compared using the measures described in the introduction. The project fabricated two synthetic datasets that

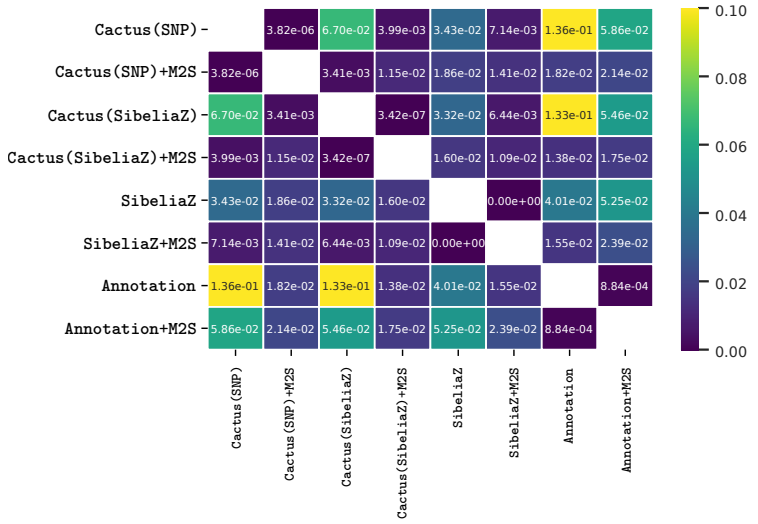

Fig. 4: The discordance ratio between each pair of block collections. Each of methods `Cactus(SNP)`, `Cactus(SibeliaZ)`, `SibeliaZ`, and Annotation along with their `maf2synteny` counterpart.

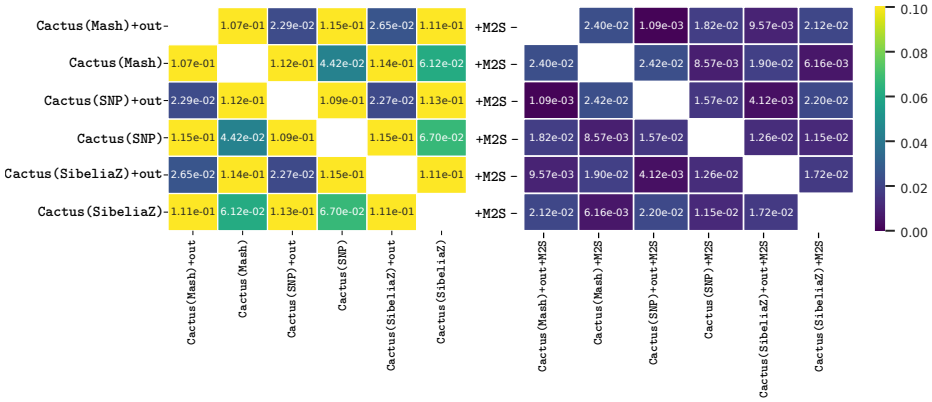

Fig. 5: The inclusion of an outgroup affects `Cactus` blocks more than the choice of a guide tree.

were used to evaluate the block collections, one that mimicked the properties of set of Primates, and another that mimicked the properties of a set of Mammals.

We applied our MDDS heuristic to each pair of block collections. Note that we were limited to the collections available on the Alignathon downloads page, so were unable to compare to some methods, such as `Mercator/Pecan` [26]. The results for the Primate dataset are depicted in Fig. 6 while results for the mammal dataset are depicted in Fig. 7. Being evolutionarily closely related, the

Table 2: Comparison of discordance ratios between blocks computed by the same method, with and without an outgroup in the input genome set. `maf2synteny` usually reduces the high discordance (shown as a percentage) between the blocks. For example, `Cactus(SNP)` applied to the TB sets with and without the outgroup shows a very high divergence ratio, yet a much lower one after `maf2synteny` has been applied to the two block collections.

maf2synteny?	Cactus(SNP)	SibeliaZ	Annotation
no	10.09%	3.57%	0.00263%
yes	1.57%	2.37%	0.0547%

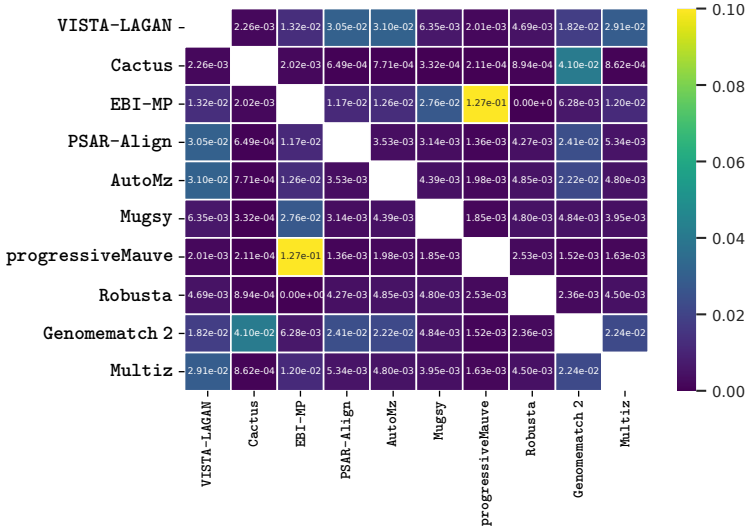

Fig. 6: Discordance ratios for simulated Primates.

Primates dataset mostly shows discordance ratios below 2%. While this trend is consistent with the Alignathon findings, including `GenomeMatch2` (SoftBerry, Mount Kisco, NY) being relatively more discordant, there were differences with the Jaccard index reported by Alignathon. `VISTA-LAGAN` [8], for instance, stands out as generally more discordant than the others, being rather dissimilar to `PSAR-Align` [16], `AutoMz`, and `Multiz` [6]. `EBI-MP` stands out as having both the best, and the worse discordance ratios of the dataset; despite having a ratio of over 12% against `progressiveMauve` [7], it also is the only method in the set to be hierarchically related to another one (`Robusta` [24]). `Cactus` has very low discordance with all methods except `GenomeMatch2`.

The simulated mammal dataset contained genomes that were separated by a larger evolutionary distance, and this was reflected in surprisingly large discordance ratios. We observe several discrepancies with the Jaccard distances reported by Alignathon ([9] – Fig. 8B). `GenomeMatch3` was extremely dissimilar

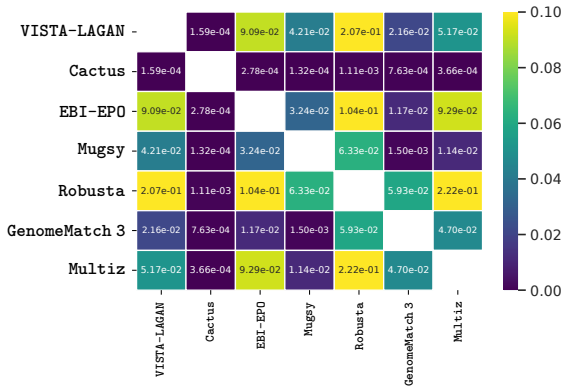

Fig. 7: Discordance ratios for simulated Mammals.

to all methods but `Mugsy` [2], yet we observe high hierarchical discordance ratios only against `Robusta` and `Multiz`. `Cactus` has low hierarchical discordance against all other methods, whereas it had high Jaccard distances against `Mugsy`, `GenomeMatch3`, and `EBI-EPO`. On the other hand, `Robusta` seemed to have poor comparisons for both the Jaccard and hierarchical measures.

5 Discussion and Conclusions

In this article we addressed the question of how to relate two collections of homology statement blocks to each other. We established a relationship between collections where we allowed overlapping parts of those collections to be hierarchically related. In the absence of these conditions, we developed a method that gives a lower bound on the number of positions that must be ignored in order for the two to be hierarchically related.

The notion of being hierarchically related depends on semantics that we imposed on the blocks, which speak to the pairwise homology relationships between the constituent genomic positions appearing in the blocks. As Ghiurcuta and Moret [11] used "homology statements" to define their "syntenic blocks", we used "positive homology witness" pairs to limit which segments can be contained within a homology statement block; while they required every homology statement within a segment to occur in all other segments of the block, we allowed positions that do not appear in a positive witness pair in the block, as long as they do not occur in another block. We went further by associating semantic meaning to the fact that two positions appear in different blocks. This allowed us to define what it means for some (sub)collection of blocks to generalize another (sub)collection.

On the algorithmic side, we showed the Minimum Deletion into Disjoint Stars problem to be NP-Complete. Our heuristic for MDDS is based on a dynamic program that solves MDDS exactly on a tree. Future improvements will include the exploration of other algorithms with provable guarantees to

the quality of their solutions. The solution to the MDDS problem gives the number of nucleotides that must be ignored so as to make the components of the block graph stars. This is a necessary condition for the two collections to be hierarchically related, but not sufficient, and thus is a lower bound on the number of nucleotides that must be ignored so as to make the two collections hierarchically related. Future work will explore ways to tighten this bound.

We studied block collections on a set of 94 *Mycobacterium tuberculosis* strains, built by annotation and non-annotation based means. We showed on this data that the agglomeration of blocks using `maf2synteny` almost always yielded collections that were less discordant. We showed surprising discordance between the gene-based annotation method and the *de novo* block inference methods `Cactus` and `SibeliaZ`. `Cactus` showed great heterogeneity, dependent on the guide tree that was used to construct the blocks.

When performing a phylogenetic analysis on the blocks, one is tempted to incorporate an outgroup for the sake of rooting the tree. We showed the inclusion of that outgroup had drastic effects on blocks, producing blocks that were less sensitive to the `Cactus` guide tree. This was concordant with our results from a phylogenetic study [10].

We studied block collections from the Alignathon project. The simulated Primates dataset showed that `EBI-MP` had both the best discordance ratio, and the worst, among all pairwise comparisons, being hierarchically related to the `Robusta` blocks while having ratio over 0.13 with `progressiveMauve`. For the less closely related simulated mammalian genomes, we showed several discrepancies between the Jaccard index reported by Alignathon and our discordance ratio, the most notable one being that while `Cactus` had a poor Jaccard index against a few methods, it had very low hierarchical discordance with all other methods (except `GenomeMatch2`).

While WGA tools and syntenic block agglomeration methods have continued to be developed, the methods to compare and analyze them has lagged behind, and the definitions of syntenic blocks are usually procedural or based on colinearity. In this article we outlined constraints on homology blocks based on the homology relationships between pairs of positions in the genome. These constraints put as much importance on the ends of the blocks as it does their contents; if two genomic segments are put into different blocks, we interpret this as a statement that should only be contradicted in a generalization of the blocks. Our new measure should inform future block inference tool development, and serve as a sanity check for the practitioner studying large scale structure of sets of genomes.

6 Availability of Code

All of the code associated with this paper is publicly availble at the following URL: https://bitbucket.org/thekswenson/homology-evaluation.

7 Acknowledgements

The authors would like to thank the reviwers for their helpful suggestions. AE and FV are supported by the NIAID grant R01AI105185.

References

1. Altenhoff, A.M., Boeckmann, B., Capella-Gutierrez, S., Dalquen, D.A., DeLuca, T., Forslund, K., Huerta-Cepas, J., Linard, B., Pereira, C., Pryszcz, L.P., et al.: Standardized benchmarking in the quest for orthologs. Nature methods **13**(5), 425–430 (2016)
2. Angiuoli, S.V., Salzberg, S.L.: Mugsy: fast multiple alignment of closely related whole genomes. Bioinformatics **27**(3), 334–342 (2011)
3. Armstrong, J., Hickey, G., Diekhans, M., Fiddes, I.T., Novak, A.M., Deran, A., Fang, Q., Xie, D., Feng, S., Stiller, J., Genereux, D., Johnson, J., Marinescu, V.D., Alföldi, J., Harris, R.S., Lindblad-Toh, K., Haussler, D., Karlsson, E., Jarvis, E.D., Zhang, G., Paten, B.: Progressive Cactus is a multiple-genome aligner for the thousand-genome era. Nature **587**(7833), 246–251 (Nov 2020), number: 7833 Publisher: Nature Publishing Group
4. Babenko, M., Gusakov, A.: New exact and approximation algorithms for the star packing problem in undirected graphs. In: Symposium on Theoretical Aspects of Computer Science (STACS2011). vol. 9, pp. 519–530 (2011)
5. Berney, M., Berney-Meyer, L., Wong, K.W., Chen, B., Chen, M., Kim, J., Wang, J., Harris, D., Parkhill, J., Chan, J., Wang, F., Jacobs, W.R.: Essential roles of methionine and S-adenosylmethionine in the autarkic lifestyle of *Mycobacterium tuberculosis*. Proceedings of the National Academy of Sciences **112**(32), 10008–10013 (Aug 2015), publisher: National Academy of Sciences Section: Biological Sciences
6. Blanchette, M., Kent, W.J., Riemer, C., Elnitski, L., Smit, A.F., Roskin, K.M., Baertsch, R., Rosenbloom, K., Clawson, H., Green, E.D., et al.: Aligning multiple genomic sequences with the threaded blockset aligner. Genome research **14**(4), 708–715 (2004)
7. Darling, A.E., Mau, B., Perna, N.T.: progressivemauve: multiple genome alignment with gene gain, loss and rearrangement. PloS one **5**(6), e11147 (2010)
8. Dubchak, I., Poliakov, A., Kislyuk, A., Brudno, M.: Multiple whole-genome alignments without a reference organism. Genome research **19**(4), 682–689 (2009)
9. Earl, D., Nguyen, N., Hickey, G., Harris, R.S., Fitzgerald, S., Beal, K., Seledtsov, I., Molodtsov, V., Raney, B.J., Clawson, H., et al.: Alignathon: a competitive assessment of whole-genome alignment methods. Genome research **24**(12), 2077–2089 (2014)
10. Elghraoui, A., Mirarab, S., Swenson, K.M., Valafar, F.: Evaluating impacts of syntenic block detection strategies on rearrangement phylogeny using *Mycobacterium tuberculosis* isolates. bioRxiv (2022)
11. Ghiurcuta, C.G., Moret, B.M.: Evaluating synteny for improved comparative studies. Bioinformatics **30**(12), i9–i18 (2014)
12. Glover, N., Dessimoz, C., Ebersberger, I., Forslund, S.K., Gabaldón, T., Huerta-Cepas, J., Martin, M.J., Muffato, M., Patricio, M., Pereira, C., et al.: Advances and applications in the quest for orthologs. Molecular biology and evolution **36**(10), 2157–2164 (2019)
13. Hell, P., Kirkpatrick, D.G.: Packings by complete bipartite graphs. SIAM Journal on Algebraic Discrete Methods **7**(2), 199–209 (1986)
14. Hu, F., Lin, Y., Tang, J.: MLGO: phylogeny reconstruction and ancestral inference from gene-order data. BMC Bioinformatics **15**(1), 354 (Nov 2014)
15. Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity? Journal of Computer and System Sciences **63**(4), 512–530 (2001)

16. Kim, J., Ma, J.: PSAR: measuring multiple sequence alignment reliability by probabilistic sampling. Nucleic acids research **39**(15), 6359–6368 (2011)
17. Kirkpatrick, D.G., Hell, P.: On the complexity of general graph factor problems. SIAM Journal on Computing **12**(3), 601–609 (1983)
18. Kolmogorov, M., Armstrong, J., Raney, B.J., Streeter, I., Dunn, M., Yang, F., Odom, D., Flicek, P., Keane, T.M., Thybert, D., Paten, B., Pham, S.: Chromosome assembly of large and complex genomes using multiple references. Genome Research **28**(11), 1720–1732 (Nov 2018)
19. Marcet-Houben, M., Gabaldón, T.: EvolClust: automated inference of evolutionary conserved gene clusters in eukaryotes. Bioinformatics **36**(4), 1265–1266 (2020)
20. Minkin, I., Medvedev, P.: Scalable multiple whole-genome alignment and locally collinear block construction with SibeliaZ. Nature Communications **11**(1), 6327 (Dec 2020), number: 1 Publisher: Nature Publishing Group
21. Modlin, S.J., Conkle-Gutierrez, D., Kim, C., Mitchell, S.N., Morrissey, C., Weinrick, B.C., Jacobs, W.R., Ramirez-Busby, S.M., Hoffner, S.E., Valafar, F.: Drivers and sites of diversity in the DNA adenine methylomes of 93 *Mycobacterium tuberculosis* complex clinical isolates. eLife **9**, e58542 (Oct 2020), publisher: eLife Sciences Publications, Ltd
22. Mural, R.J., Adams, M.D., Myers, E.W., Smith, H.O., Miklos, G.L.G., Wides, R., Halpern, A., Li, P.W., Sutton, G.G., Nadeau, J., et al.: A comparison of whole-genome shotgun-derived mouse chromosome 16 and the human genome. Science **296**(5573), 1661–1671 (2002)
23. Nadeau, J.H., Taylor, B.A.: Lengths of chromosomal segments conserved since divergence of man and mouse. Proceedings of the National Academy of Sciences **81**(3), 814–818 (1984)
24. Notredame, C.: Robusta: a meta-multiple genome alignment tool (2012), `http://www.tcoffee.org/Projects/robusta`
25. Ondov, B.D., Treangen, T.J., Melsted, P., Mallonee, A.B., Bergman, N.H., Koren, S., Phillippy, A.M.: Mash: fast genome and metagenome distance estimation using MinHash. Genome Biology **17**(1), 132 (Jun 2016)
26. Paten, B., Herrero, J., Beal, K., Fitzgerald, S., Birney, E.: Enredo and pecan: genome-wide mammalian consistency-based multiple alignment with paralogs. Genome research **18**(11), 1814–1828 (2008)
27. Pevzner, P., Tesler, G.: Genome rearrangements in mammalian evolution: lessons from human and mouse genomes. Genome research **13**(1), 37–45 (2003)
28. Prakash, A., Tompa, M.: Measuring the accuracy of genome-size multiple alignments. Genome biology **8**(6), 1–11 (2007)
29. Sturtevant, A.H., Novitski, E.: The homologies of the chromosome elements in the genus drosophila. Genetics **26**(5), 517 (1941)
30. Svetlitsky, D., Dagan, T., Ziv-Ukelson, M.: Discovery of multi-operon colinear syntenic blocks in microbial genomes. Bioinformatics **36**(Supplement_1), i21–i29 (2020)
31. Waterston, R.H., Lindblad-Toh, K., Birney, E., Rogers, J., Abril, J.F., Agarwal, P., Agarwala, R., Ainscough, R., Alexandersson, M., An, P., et al.: Initial sequencing and comparative analysis of the mouse genome. Nature **420**(6915), 520–562 (2002)
32. Weller, M., Komusiewicz, C., Niedermeier, R., Uhlmann, J.: On making directed graphs transitive. J. Comput. Syst. Sci. **78**(2), 559–574 (2012)
33. Winter, S., Jahn, K., Wehner, S., Kuchenbecker, L., Marz, M., Stoye, J., Böcker, S.: Finding approximate gene clusters with gecko 3. Nucleic acids research **44**(20), 9600–9610 (2016)

A NP-Hardness of MDDS

Note that the notion of star and *induced* star coincide on bipartite graphs since, for any bipartite G, the vertices of any star-subgraph of G also form an induced star in G. Further, no collection of node-disjoint stars can contain the triangle C_3 or the path on 4 vertices P_4 as a subgraph and it can be seen that this condition is also sufficient.

Observation 1 *A bipartite graph G is a collection of stars if and only if G does not contain a P_4 subgraph.*

For the correctness proof, we will make two assumptions on the structure of the input formula φ, without loss of generality. First, we assume that no variable occurs in all clauses. If a variable x does occur in all clauses, then we simply add a new variable y and the singleton clause on y. Second, we assume that each clause in φ has exactly three literals. If a clause C has at most two literals, we can simply double the occurrence of any literal in C.

Lemma 2. *Let φ be an instance of* 3SAT *and let $(G = (V, E), \omega)$ be the result of applying Construction 1 to φ. Then, φ is satisfiable if and only if (G, ω) has a star packing of weight at least $12m^2 + m$.*

Proof. For each variable x_i of φ, let us define the edge sets

$$T_i := \bigcup_{0 \le j < 2n_i} \{v_i^{3j} v_i^{3j+1}, v_i^{3j+2} v_i^{3j \oplus 3}\} \quad \text{and} \qquad F_i := \bigcup_{0 \le j < 2n_i} \{v_i^{3j+1} v_i^{3j+2}, v_i^{3j+2} v_i^{3j \oplus 3}\}$$

where $3j \oplus 3 := (3j + 3) \bmod 6n_i$. Note that any v_i^j has degree two in subgraph (V, T_i) if and only if $j \equiv 0 \mod 3$ and any v_i^j has degree two in (V, F_i) if and only if $j \equiv 2 \mod 3$. Further, $\omega(T_i) = \omega(F_i) = 4mn_i$. We prove the two directions of the lemma separately.

$\Rightarrow$: Let φ be satisfiable, that is, there is a set L of literals over variables in φ such that each clause C_k intersects L in at least one literal ℓ_k and L contains exactly one of x_i and $\neg x_i$ for all i. If ℓ_k is the literal x_i in clause $C_k = \gamma_i^j$, then let $e_k := u_k v_i^{6j}$ and, if ℓ_k is the literal $\neg x_i$, then let $e_k := u_k v_i^{6j+2}$. Note that all e_k are distinct, $e_k \in E(G)$ for all k. Let S^{clause} contain e_k for all clauses C_k of φ and note that $\omega(S^{\text{clause}}) = m$. Further, for all variables x_i of φ, let $S_i^{\text{var}} := T_i$ if $x_i \in L$ and $S_i^{\text{var}} := F_i$, otherwise (that is, $\neg x_i \in L$). Finally, let the selected edges be $S := S^{\text{clause}} \cup \bigcup_i S_i^{\text{var}}$ (see gray edges in Fig. 3), noting that $\omega(S) = 12m^2 + m$. It remains to show that (V, S) does not contain a P_4 as a subgraph. Towards a contradiction, assume that (V, S) contains a P_4 $p := (a, b, c, d)$. By construction, neither $\bigcup_i S_i^{\text{var}}$ nor S^{clause} contains a P_4, and p must contain edges from both of these sets. Thus, p contains u_k for some clause C_k. Since any u_k has degree one in (V, S), we can assume without loss of generality that $ab = e_k$. By definition of e_k, there are i and j such that either $b = v_i^{6j}$ and $x_i \in L \cap C_k$ or $b = v_i^{6j+2}$ and $\neg x_i \in L \cap C_k$. Since $6j \equiv 0 \bmod 3$ and $6j + 2 \equiv 2 \bmod 3$ we know that in both

cases b has degree two in (V, S_i^{var}), and both of its neighbors have degree one in (V, S_i^{var}) and, thus, in (V, S). This contradicts (a, b, c, d) being a path in (V, S).

$\Leftarrow$: Let S be a maximum-weight subset of E such that $\omega(S) \geq 12m^2 + m$ and (V, S) does not contain a P_4 as a subgraph. First, let S^{clause} denote the set of edges of S incident with a clause node u_k. Second, for each x_i, let S_i^{var} denote the set of edges of S on the variable cycle corresponding to x_i and note that, for each P_3 (a, b, c) in (V, S_i^{var}), both a and c have degree one in (V, S_i^{var}) since, otherwise, (V, S) contains a P_4. For each i, the connected components of (V, S_i^{var}) are paths of lengths 1, 2, or 3 and we denote the number of P_1s, P_2s, and P_3s in (V, S_i^{var}) by r_i, s_i and t_i, respectively. By construction, each P_2 is adjacent to at most one clause vertex u_k in (V, S), and since (V, S) does not contain P_4, each P_3 is also adjacent to at most one clause vertex u_k in (V, S).

Claim. $\sum_i t_i = 6m$.

Proof. By decomposing the $18m$ vertices of the variable cycles into P_3 subgraphs separated by single edges, the upper bound of $6m$ is attained. It suffices to show $\sum_i t_i \geq 6m$ so, towards a contradiction, assume that $\sum_i t_i < 6m$. Then, there is a variable x_i such that $t_i < 2n_i$ implying $|S_i^{\text{var}}| \leq 4n_i - 1$ by construction. Let S' result from S by removing all edges incident with vertices of the variable cycle corresponding to x_i and adding the edges in T_i. Since x_i does not occur in all clauses, we removed edges of total weight strictly less than $m + (4n_i - 1)m = 4mn_i$ and we added edges of total weight $m|T_i| = 4mn_i$. Since neither (V, S) nor (V, T_i) contains a P_4, neither does (V, S'), thus contradicting optimality of S. ■

Corollary 1. *Each subgraph (V, S_i^{var}) decomposes into disjoint copies of P_3.*

Corollary 2. *Let v_i^j and $v_i^{j'}$ be nodes of degree two in some subgraph (V, S_i^{var}). Then, $|j - j'| \equiv 0 \mod 3$.*

Corollary 3. *Let $u_k v_i^j$ be an edge in S^{clause}. Then, v_i^j has degree two in (V, S_i^{var}).*

Note that each P_3 in $(V, \bigcup_i S_i^{\text{var}})$ has weight exactly $2m$, so S contains exactly m edges of S^{clause}. Further, by Corollary 3, all clause vertices u_k have degree at most one since they are not adjacent to degree-one vertices. Together, this means that all clause vertices u_k are incident to *exactly* one edge in S.

We now construct an assignment β and show that it satisfies φ. To this end, let $\beta(x_i) = \text{TRUE}$ if and only if S contains the edge $u_k v_i^{6j}$ for some $j, k \in \mathbb{N}$. Note that, if S contains the edge $u_k v_i^{6j}$ for any $j, k \in \mathbb{N}$ then, by Corollary 1, v_i^{6j} has degree two in (V, S_i^{var}). Then, by Corollary 2, S cannot contain the edge $u_{k'} v_i^{6j+2}$ for any $j', k' \in \mathbb{N}$. Thus, β is well-defined. It remains to show that β satisfies φ. To this end, let C_k be any clause in φ, let $u_k z$ be the unique edge incident with u_k in S and let x_i be the variable whose variable cycle contains z. If x_i occurs non-negated in C_k, then $z = v_i^{6j}$ for some $j \in \mathbb{N}$ by construction. But then, $\beta(x_i) = \text{TRUE}$ and x_i satisfies C_k. If x_i occurs negated in C_k, then $z = v_i^{6j+2}$ for some $j \in \mathbb{N}$ by construction. But then, $\beta(x_i) = \text{FALSE}$ and x_i satisfies C_k. In both cases, C_k is satisfied. □

B Collections of Block that are not Clean

Many of the software we studied produced blocks that were not clean, containing blocks with overlapping segments. We removed overlapping segments by visiting pairs of blocks in an arbitrary order, removing the overlap between their overlapping segments. Although the order in which overlaps are removed can effect the final set of blocks, we made the process deterministic by visiting the pairs in a fixed order.

C Segmental Duplications

If some method makes orthology predictions that may contain multiple segments from the same genome (*e.g.* clusters of orthologous groups that contain paralogs from a single genome), the block graph may provide insight into how to refine the orthology groups using blocks from another method. This section outlines such a case.

When a block $A \in \mathcal{A}$ contains multiple segments from multiple genomes, blocks from another set $B1, B2 \in \mathcal{B}$ could overlap in ways that create non-star graph topologies. Fig. 8 shows one such example.

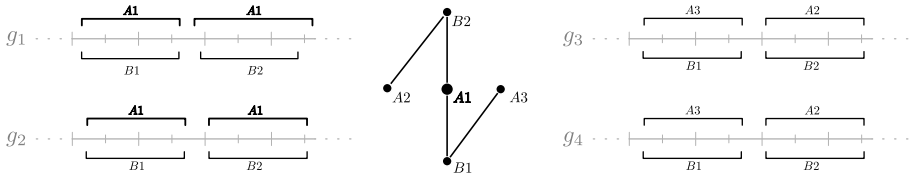

Fig. 8: The block $A1 \in \mathcal{A}$ has two (duplicated) segments in genomes g_1 and g_2. The blocks $B1, B2 \in \mathcal{B}$ each overlap with one of the two copies. This configuration creates the non-star topology depicted in the middle. The block $A1$ can easily be split into two so that the graph becomes only stars. This results in a refinement of the blocks of $\mathcal{A}$, based on the blocks of $\mathcal{B}$.

Blocks $B1$ and $B2$ each overlap one of the two duplicate copies of $A1$ in genomes g_1 and g_2. The block $A1$ can be split into two blocks $A1'$ and $A1''$ such that the collections $\{A1', A1'', A2, A3\}$ and $\mathcal{B}$ are hierarchically related. The two connected components of $BG(\{A1', A1'', A2, A3\}, \mathcal{B})$ are both stars with vertices $\{A2, B2, A1'\}$ and $\{A1'', B1, A3\}$. This transformation can be generalized to vertices of higher degree, as long as the overlapping segments can be split in this way.