Optimizing Directed Acyclic Graphs via Simulated Annealing for Reconstructing Human Segmental Duplications

Borislav H. Hristov
Department of Computer Science, Brown University, Providence, RI, USA
bhristov@cs.brown.edu

May 1, 2010

Abstract

Segmental duplications, relatively long and nearly identical regions, prevalent in the mammalian genome, are successfully modeled by directed acyclic graphs. Reconstructing the evolutionary history of these genomic regions is a non-trivial, but important task, as segmental duplications harbor recent primate-specific and human-specific innovations and also mediate copy number variation within the human population. Using novel models derived by Kahn and Raphael, we formalize this reconstructon task as an optimization problem on the space of directed acyclic graphs. We employ a simulated annealing heuristic and describe an efficient way to use the technique to solve the optimization problem in general. We apply the heuristic to both maximum parsimony and maximum likelihood evolutionary models. We use these models to analyze segmental duplications in the human genome and reveal subtle relationships between these blocks.

1 Introduction

Graphs in general, and directed acyclic graphs in particular, are powerful tools for modeling and studying a huge variety of problems. Graphs have been the subject of intense research in mathematics and computer science, and numerous algorithms for computing many of their properties have been developed. However, because the space of graphs grows super exponentially with the number of vertices, a broad range of problems (such as Travelling Salesman, Complete Coloring, Minimum Cut) are NP hard and only approximation algorithms are available. Among those computationally hard tasks is the problem of finding optimal directed acyclic graph (DAG) over the space of DAGs with respect to a given graph metric (we will define one such problem rigorously in a later section). Here, we describe an efficient way to utilize simulated annealing technique to search the space of DAGs. A key component in our strategy is the ability to efficently move from one location to another which we achieve via incidence and ancestor matrices. We examine the main properties of a simulated annealing algorithm and implement a generic approach to optimize over the space of DAGs.

One of the many applications of DAGs is modeling evolutionary history between biological entities when ancestral relations are represented as directed edges and an entity has more than one direct ancestor. Of particular interest to us are segmental duplications because they harbor recent primate-specific and human-specific innovations and also mediate copy number variation within the human population. Moreover, segmental duplications account for a significant fraction of the differences between humans and other primate genomes, and are enriched for genes that are differentially expressed between the species [6]. Reconstructing the evolutionary history of these genomic regions is an important task which remains an extreme challenge, as they are the "most structurally complex and dynamic regions of the human genome" [1].

In [16, 17] Kahn and Raphael introduce a novel measure called "duplication distance". This measure evaluates the similiarity between strings by counting the minimum number of "copy" operations needed to generate one string from the other, i.e the maximum parsimony scenario. In [14] we present a novel probabilistic model of segmental duplication that we use to compute the likelihood score for an evolutionary relationship between a pair of duplication blocks, i.e the maximum likelihood scenario.

Using these two models, we formalize the reconstruction of the evolutionary history of segmental duplication as a problem of finding optimal directed acyclic graph. We employ simulated annealing to solve for the maximum parsimony and maximum likelihood reconstructions and compare them to the analysis of [13]. Our evolutionary reconstruction recapitulates some of the properties of earlier analysis but also reveals additional and more subtle relationships between segmental duplications. A more comprehensive exposition of our analysis can also be found in [14].

2 Biological Background

Segmental duplications are relatively long, nearly identical regions of the genome. Bailey and Eichler [4] find that approximately 5% of the human genome consists of segmental duplications > 1 kb in length and with $\geq 90\%$ sequence identity between copies. Interestingly, segmental duplications often form complex mosaics of contiguous regions known as duplication blocks. Human segmental duplications contain novel fusion genes [27], genes under strong positive selection [8], and new gene families [20]. Moreover, the presence of segmental duplications appears to render regions of the genome more susceptible to recurrent and disease-causing rearrangements [21] as

well as additional copy-number variants [5] and inversions [28]. Since segmental duplications arise as copy-number variants that become fixed in a population, the evolutionary history of segmental duplications reveals information about the mechanisms and temporal dynamics of copy-number variants in the human genome [18].

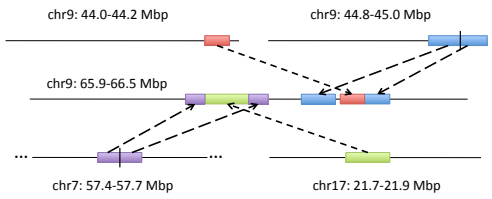

Figure 1: Example of duplicate events leading to the construction of duplication block *chr9:65.9-66.5Mbp*

Jiang et al. [13] produced a comprehensive annotation of this mosaic organization; they derived an "alphabet" of approximately 11,000 duplicated segments, or **duplicons**, and delimited 437 **duplication blocks**, or "strings" of at least 10 (and typically dozens) different duplicons found contiguously on a chromosome. However, the relationships between these annotated duplication blocks are complex and straightforward analysis does not immediately reveal the evolutionary relationships between blocks.

2.1 Parsimonious Model

Kahn and Raphael [16, 17] introduce a novel parsimonious model of segmental duplications based on duplication distance. They define *duplicate operation* as the basic operation $\delta_{s,t,p}(X)$ which copies a substring $X_{s,t}$ of a source string X and pastes it into a target string at position p. The *duplication distance*, $d(X, Y)$, for a source string X and a target string Y is then defined to be the minimum number of duplicate operations needed to construct Y by copying and pasting substrings of X into an initially empty target string. In [15], Kahn and Raphael cleverly develop a polynomial-time algorithm to compute $d(X, Y)$. We note that the duplication distance is not formally a distance because it is asymmetric, $d(X, Y) \neq d(Y, X)$.

2.2 Likelihood Model

While the parsimonious model is attractive from a theoretical perspective and can produce useful biological insight, it might be overly restrictive, particularly when there are many different optimal or nearly optimal solutions. In [14] we define for a given source string X, a *feasible generator* $\Phi_X = (X_{i_1,j_1}, \ldots, X_{i_k,j_k})$ to be a sequence of substrings of X such that:

1. the elements of Φ_X partition the characters of Y into mutually non-overlapping subsequences $\{S_1, \ldots, S_k\}$,
2. there exists a bijective mapping $f : \{X_{i,j} \in \Phi_X\} \rightarrow \{S_1, \ldots, S_k\}$ from substrings of X to subsequences in Y corresponding to how the elements of Φ_X partition Y, and
3. the order of elements in Φ_X corresponds to the order of the leftmost characters of the subsequences $f(X_{i_1,j_1}), \ldots, f(X_{i_k,j_k})$ in Y.

A sequence of duplicate operations that constructs Y from X in k operations uniquely defines a feasible generator Φ_X with length k whose elements correspond, respectively, to substrings of X that are duplicated conjointly in a single operation. Now, consider, as in [14], the following example:
$X = abcdefghijkl$ and $Y = \mathbf{a}gd\mathbf{b}he\mathbf{c}if\mathbf{d}aj\mathbf{e}bk\mathbf{f}cl\mathbf{g}$
The duplication distance, $d(X, Y)$, is 13 and there is a single feasible generator with this optimum length. However, there are 989 possible feasible generators for Y, 119 of which have length 14, just slightly suboptimal.

Because the space of all possible feasible generators is very large, a probabilistic model might give very low probability to an optimal parsimony solution. Thus, it is more enlightening to consider a probabilistic model.

In [14] we introduce a novel likelihood score which is derived by computing the weighted ensemble of all possible duplication scenarios.

3 Optimizing DAGs via Simulated Annealing

In this section we define formally the problem of optimizing DAGs over the space of DAGs:

Definition 1. *Let $\mathcal{G}$ be the set of directed acyclic graphs (DAGs) with n vertices. Let $\Psi : \mathcal{G} \rightarrow \mathbb{R}$ be a real valued function over $\mathcal{G}$. The optimal graph is* $\operatorname{argmin}_{G \in \mathcal{G}} \Psi(G)$.

By a reduction from the problem of Learning Bayesian Networks, this problem is NP-hard [7]. Thus, we use Simulated Annealing approach to derive approximate solutions.

3.1 Simulated Annealing

Simulated Annealing is a powerful heuristic technique for global optimization problems. While it is not guaranteed that the method will find *the optimum* it has been shown to efficiently find close solutions even in the presence of noisy data. The algorithm was described by Kirkpatrickis et al in [19]. It is based upon that of Metropolis et al. [22] which generates sample states of a thermodynamic system. Simulated Annealing is named for its similiarity to the metalurgy process of annealing in which metal is first heated and then gradually cooled so that its atoms could settle at optimal crystalline structure and thus give the metal more strength. To continue the analogy, in the algorithm we initially allow for moves that increase the objective function (heating phase) but as the time progresses we accept only moves that improve the objective (cooling phase). More formally, the algorithm performs a random search and accepts changes with probability $p = exp(\frac{-\Delta E}{T})$ where ΔE is the change in the energy (objective function) and T is temperature parameter which is decreased according to a given cooling schedule. Clearly, when the temperature parameter is high, bad moves are more likely to be accepted; when $T = 0$ the probability of accepting such moves goes to zero. Allowing for those bad moves provides the key advantage of Simulated Annealing over any greedy algorithm- the ability to avoid being trapped in local optima.
In order to implement an efficient Simulated Annealing algorithm for the space of DAGs we need to be able to efficiently explore this space, evaluate the change of the objective function, and decide whether move in that direction at all. Next we describe how to do so.

3.2 Exploring the space of DAGs

In [11] Giudici and Castelo describe an elegant approach for moving from one DAG to another via three types of simple moves: *adding* a new edge, *removing* an existing one, or *reversing* an existing one. Clearly, addition and removal are sufficient as reversal is easlily mimicked by performing appropriate addition and removal. However, as illustrated in [11] there is an important consideration to allow reversals in order to achieve faster convergence. Imagine that the configuration $a \rightarrow b$ is the most probable one. If, during random steps of the exploration phase, the edge $a \leftarrow b$ of opposite direction is chosen, it might be very difficult to remove it later because of the strong marginal dependence between a and b. This is not an issue if we include reversals.

It is clear that these moves result in a directed graph. The key question is whether they preserve the acyclic property of the graph. We shall say that a move is *legal* provided the resulting directed

graph is acyclic again. An efficient way of testing whether a move is legal is by using the incidence and the ancestory matrices [11]. Let us formally define:

Definition 2. *Let $G = (V, E)$ be a DAG with set of n vertices V and set of edges E.*
The **Incidence** *matrix of G is $n \times n$ matrix I with entries $I(i, j) = 1$ if and only if there is a directed edge $v_j \rightarrow v_i$ in G and $I(i, j) = 0$ otherwise.*
The **Ancestory** *matrix of G is $n \times n$ matrix A with entries $A(i, j) = 1$ if and only if there is a directed path from v_j to v_i in G (i.e. v_j is ancestor of v_i) and 0 otherwise.*

Definition 3. *Given DAG $G = (V, E)$ we call the DAG $G' = (V, E')$ a* **neighbor** *of G if and only if we can obtain G' from G with a single move- adding a new edge, removing, or reversing an existing edge in G.*

Definition 4. *Given an objective function Ψ and two DAGs G_1, G_2 we call $\Delta G = \Psi(G_1) - \Psi(G_2)$* **the difference in their energies**.

Now, given a DAG $G = (V, E)$ and a random move proposed by the Simulated Annealing we need to:
1. Examine whether the move is legal
2. Decide wheter to accept the move based on the probability $p = exp(\frac{-\Delta G}{T})$
3. Perform the move
Therefore, we split the necessary computational operations in these three phases.

3.3 Legal Moves

When we have to decide whether a proposed move introduces a directed cycle or not we need to examine the three possible moves separately:

- *Addition.* Consider adding an edge from v_i to v_j. It is sufficient to look at the ancestory matrix. If $A(i, j) = 0$, then the move is legal and therefore permitted. If $A(i, j) = 1$, then v_j is ancestor of v_i and thus adding the edge $v_i \rightarrow v_j$ would introduce a directed cycle. Examining the matrix is $O(1)$.

- *Removal.* Clearly, removing an edge could never create a cycle and therefore is always legal, $O(1)$.

- *Reversal.* We can analyze the reversal of an edge $v_i \rightarrow v_j$ as a two step move: first removing the edge $v_i \rightarrow v_j$ (always legal) and then adding the edge $v_j \rightarrow v_i$. However, this time checking only the value of $A(i, j)$ is not enough. As there might be another directed path from v_i to v_j we need to examine every vertex v_k that is in the ancestorship of v_i and is parent of v_j and look at the value $A(k, j)$. This takes $O(n)$ time.

Therefore, to decide whether a move is legal it takes $O(1)$ for addition and removal, and $O(n)$ for reversal. We note that we use $O(n^2)$ space to store the matrices I and A.

3.4 Accepting a Move

To decide whether to accept a move or not we need to compute $p = exp(\frac{-\Delta G}{T})$. Then, we compare p with a random number in the interval $(0, 1)$ and if $p > rand(0, 1)$ we accept the move. We note that depending on the complexty of the objective function $f(G)$ computing ΔG could be very expensive. In fact, this is the case for the max likelihood reconstruction because computing $Pr[Y|X, k]$ takes in the worst-case $O(|Y|^3|X|k^2)$. Therefore, we employ a hashtable to store the cost of every move we have examined. As we do hundreds of independent trials we may often need to examine the same move multiple times, and the hashtable helps significantly speed up the search for good moves.

3.5 Performing a Move

Once a move is accepted we update the incidence and ancestor matrix to perform it. Again, we consider the three types of moves separetely:

- *Addition.* Suppose we add $v_i \rightarrow v_j$. We need only to set $I(j, i) = 1$ to update the incidence matrix. For the ancestor matrix, the first step is to set all ancestors of v_i as ancestors of v_j. The second is to add to the ancestors of the descendants of v_j the ancestors of v_j. Because we keep the necessary information in the ancestor matrix A we need to do this for every $k = 1, 2, ..., n$ for which $A(k, j) = 1$. The complexity is $O(n)$.

- *Removal.* Let $v_i \rightarrow v_j$ be the edge to remove. In this case, the very first thing is to set $I(j, i) = 0$. Then, for v_j and all its descendant we need to rebuild the corresponding rows of A. We start with v_j by first setting his ancestors to be all of his parents, and then we add to v_j's ancestors all ancestors of his parents. We repeat this procedure with all descendants of v_j in topological order. The complexity is $O(n^2)$.

- *Reversal.* Let $v_i \rightarrow v_j$ be the edge to reverse. This is done by first removing $v_i \rightarrow v_j$ and then by adding $v_j \rightarrow v_i$. The complexity is $O(n)$.

We observe that it takes $O(n^2)$ time for removing and reversing but those moves are performed only after they are accepted. This is another advantage of this approach as costly opperations are done only when needed.

3.6 Cooling Schedule

Determining what the cooling schedule should be has been a question of great interest and extensive research (see [23] and [3]). One of the most commonly used schedules is the exponential, originally introduced in [19]. The temperature is updated via the equation $T_{t+1} = T_t\alpha$. Another widely used is the linear $T_{t+1} = T_t - \mu$. Of particular theoretical importance is the logarithmic cooling scheme developed by Geman and Geman [10]: $T_t = \frac{c}{log(t+d)}$, where d is constant, ususally set to one. Hajek proves in [12] that if c is greater then the biggest energy barrier, given infinite time the algorithm will converge to the global optima. However, this asymptotic decrease in temperature is so slow that it is impractical to use in any real situation (it might be faster to do exhaustive search of the space) and it stands as a theoretical result.

Another cooling schedule is *threshold acceptance* introduced by Dueck and Scheuer [9]. In threshold acceptance, bad moves are accepted if the increase in the energy is less than a fixed threshold, and good moves are always accepted. The threshold is gradually lowered with each

step. This strategy has the advantage of being less computationally expensive than the exponential one as it doesn't need to compute exponentials.

In our implementation we employ the exponential schedule, though we allow for any cooling schedule to be used (even one externally specified). After testing different values for the problem of reconstructing human segmental duplications we find $\alpha = 0.98$ to perform best in terms of efficiency and time. We also find that threshold acceptence perform slightly better in some cases but only at the cost of greatly increased computational time making it impractical.

4 Exerimental Results - Reconstruction of Human Segmental Duplications

In this section, we formalize the problem of computing a segmental duplication evolutionary history for a set of duplication blocks in the human genome with respect to either a parsimony or likelihood criterion. Note that this material is adapted from the manuscript of [14].

The input to the problem is the set of duplication blocks found in the human genome, each represented as a signed string on the alphabet of duplicons. Our goal is to compute a putative duplication history that accounts for the construction of all of the duplication blocks starting from an ancestral genome that is devoid of segmental duplications. A duplication history is a sequence of duplicate events that first builds up a set of *seed* duplication blocks by duplicating and aggregating duplicons from their ancestral loci and then successively construct the remaining duplication blocks by duplicating substrings of previously constructed blocks.

We make several simplifying assumptions. First, we assume that only duplicate events occur and that there are no deletions, inversions, or other types of rearrangements within a duplication block. Second, we assume that a duplication block is not copied and used to make another duplication block until *after* it has been fully constructed. As a result of the second assumption, the set of duplication blocks observed at the present time includes both recently created blocks as well as "fossils" of seed blocks that were duplicated to construct other blocks. In [25, 24], the authors make a similar assumption when deriving the evolutionary tree for Alu and other (retro)transposons. They define the ancestral relations between mobile elements as the minimum spanning tree with respect to particular distance metric. Building an ancestry tree for duplication blocks, however, is not appropriate as duplication blocks can have multiple parent blocks.

A more appropriate description of the ancestral relationships between duplication blocks is a directed acyclic graph (DAG). In an ancestry DAG, the vertices represent duplication blocks and an edge directed from a vertex u to a vertex v indicates that u is a parent of v. A vertex with multiple incoming edges and, therefore, multiple parents, is constructed using substrings of all of the parent blocks. Specifically, given a DAG $G = (V, E)$ and $v \in V$, we define $P_G(v)$, the parent string of v, by $P_G(v) = v_1 \odot v_2 \odot \cdots \odot v_p$ where $v_i \in \{v|(v, Y) \in E\}$ and $\odot$ indicates the concatenation of two strings with a dummy character inserted in between. Our second assumption – that a duplication block can only be a parent to another block once it has been fully completed – gives the condition that the ancestral relationships cannot contain cycles. We acknowledge that our two simplifying assumptions restrict the evolutionary history reconstruction problem significantly, but admit an efficient and consistent method of scoring a solution.

4.1 Maximum Parsimony Reconstruction

We define the optimal DAG with respect to a parsimony criterion using duplication distance (Sec. 2.1 and [14]).

Definition 5. *Given a set of duplication blocks $\mathcal{D}$, the **maximum parsimony evolutionary history** is the DAG $G = (\mathcal{D}, E)$ that minimizes $\Psi(G) = \sum_{Y \in \mathcal{D}} d(P_G(Y), Y)$.*

We use a simulated annealing heuristic described in Section 3 to compute the maximum parsimony evolutionary history for a set $\mathcal{D}$ of 391 duplication blocks identified by [13]. The resulting DAG (Supplementary Figure 7) contains 391 nodes and 479 edges. There are 9 connected components with at least 4 nodes, and nearly 40% of the nodes appear in the largest connected component. Figure 6 shows a moderately-sized connected component. Note that there are long, directed paths, for example the path (383, 141, 250, 118), where each block is descended from a single parent. There are also some "hub" nodes with high out-degree, such as 395, 399, and 267. These blocks are suggestive of the *primary duplication blocks* that seed multiple other duplication blocks in the two-step model of segmental duplication (reviewed in [5]). In total, the graph contains 28 nodes with no incoming edges in all the components containing at least 4 nodes. Conversely, 89 nodes in the graph have multiple parents suggesting that they were likely the result of more complicated rearrangements where duplicons from disparate loci were copied and aggregated into a contiguous duplication blocks. The graph also contains a total of 105 singleton nodes for which we did not infer any ancestral relations with other duplication blocks, mostly (97 nodes) due to our restriction in the longest common subsequence.

The maximum parsimony DAG represents a scenario in which all 391 duplication blocks could have been constructed in a sequence of 17,431 total duplicate operations. As a baseline comparison, a minimum spanning tree, with respect to duplication distance, on the set of duplication blocks has a total parsimony score of 28,852 and, by definition, contains 390 edges. It is notable that the total score of the MST, is significantly worse than that of any graph obtained via simulated annealing. This suggests that the ancestor relationships between duplication blocks could not be captured by simple analysis (as also noted in [16]).

4.2 Maximum Likelihood Reconstruction

We can also define the optimal DAG with respect to a likelihood criterion. In phylogenetic tree reconstruction, a maximum likelihood solution is a tree that maximizes the probability of generating the characters at the leaf nodes over all possible tree topologies, branch lengths, and assignments of ancestral states to the internal nodes. Typically, the evolutionary process is assumed to be a Markov process so that the probabilities along different branches are independent. We similarly define the maximum likelihood DAG using the probabilistic model derived in [14], where the "branch lengths" are replaced by the number of duplicate operations used to construct a target block from all of its parents.

Definition 6. *Given a set of duplication blocks $\mathcal{D}$, the **maximum likelihood evolutionary history** is the DAG $G = (\mathcal{D}, E)$ that maximizes the likelihood:*

$$\begin{aligned} L(G) &= \prod_{Y \in \mathcal{D}} L(Y), \\ &= \prod_{Y \in \mathcal{D}} \left(\max_k Pr[F|Y, P_G(Y), k] \right), \\ &= \prod_{Y \in \mathcal{D}} \left(\max_k Q^{(k)}_{P_G(Y)}(Y) / Z^{(k)}_{P_G(Y)} \right), \end{aligned}$$

where $Z^{(k)}_{P_G(X)}$ and $Q^{(k)}_{P_G(Y)}$ are the partition function and restricted partition functions, defined in [14].

We computed the maximum likelihood DAGs for the sets of duplication blocks appearing within moderately-sized connected components of the maximum parsimony DAG in order to compare the two methods. We chose the components comprised of blocks from clades 'chr16' and 'chr10', respectively. The maximum likelihood subgraphs for these subproblems are shown in Fig. 3(b) and 2(b).

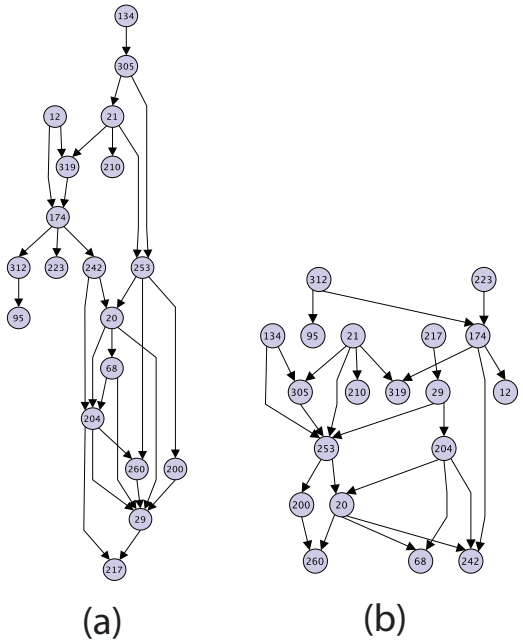

Figure 2: (a) Component comprised entirely of duplication blocks from clade 'chr10' in the maximum parsimony DAG. (b) Maximum likelihood DAG for subgraph induced on nodes in (a).

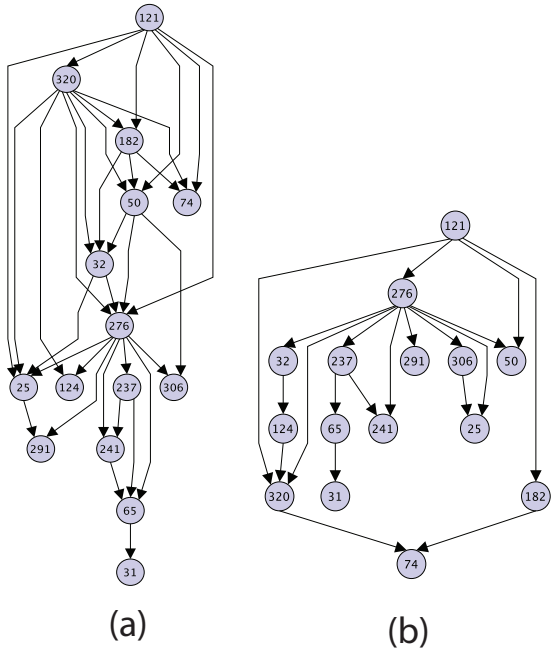

Figure 3: (a) Component comprised entirely of duplication blocks from clade 'chr16' in the maximum parsimony DAG. (b) Maximum likelihood DAG for subgraph induced on nodes in (a).

We note that an optimal DAG with respect to either parsimony or likelihood is intended only to represent an approximation of a true duplication history for a set of duplication blocks. Lacking a definitive and comprehensive way to trace the evolutionary history of human segmental duplications, we propose these two problems as a computational means for deriving a likely duplication history for duplication blocks in order to gain insight into their relationships in the same way that max parsimony and max likelihood phylogenies have been predicted for gene families.

4.3 Comparison between Maximum Parsimony and Maximum Likelihood Reconstructions

The two DAGs for the 'chr16' subproblem in Fig. 3 share some characteristics. For example, node 121 is a common ancestor of every other block and block 276 exhibits high out-degree in both solutions. But overall, the max likelihood DAG is considerably different than the max parsimony DAG. One striking difference is the higher average in-degree for blocks in the parsimony solution (2.2) as compared to the likelihood solution (1.3). A possible explanation is that the parsimony score for a block does not decrease with arbitrarily many parent blocks. Another notable difference is the greater length of the longest path in the parsimony graph (10) as opposed to the likelihood graph (6) in Fig. 3. The parsimony graph in Fig. 2 also exhibits a longer path (12)

than the longest path in the likelihood graph (5). The consequence is that the duplication histories represented by the parsimony graphs require more "generations" of duplication block construction than the likelihood graphs that represent duplication histories in which more blocks could have been constructed contemporaneously.

A very interesing comparison comes from computing the parsimony score of the maximum likelihood solution and the likelihood score of the maximum parsimony solution. The maximum likelihood solution has a very good parsimony score, close to the optimal one. This holds as a general rule comparing optimal DAGs from different clades. However, the reverse is not true. Often the maximum parsimony solution is given extremely low probability. For example, the optimal DAG from Fig. 3 (a) has a very low probability. It is caused by the edge 50→ 32. The node 32 has the set of parents [320,182,50] and this gives the most parsimony score. Removing the edge 50→ 32 and reducing the set of parents to [320,182] worsens slightly the parsimony score (by three points) but does greatly increase the likelihood. This is example shows on real data how probabilistic model gives very low probability to an optimal parsimony solution as we theoretically discussed in Section 2.3.

Before further examining the biological meaning of the derived maximum parsimony and maximum likelihood reconstructions we take a closer look at some of the properties of our simulated annealing to verify the correctness of the approach and evaluate its robustness.

4.4 Independence of the Starting Location

A good way to evaluate the robustness of the simulated annealing techique is to examine the scores it achieves starting from different initial conditions; for our problem, whether the optimal score achieved depends on the initial directed acyclic graph the annealing started from. If we obtain nearly identical optima regardless of the starting location we would feel more confident about the robustness of our heuristic. To test this we used two subsets of nodes: those corresponding to the duplication blocks labeled in [13] as clade *chr10* and those labeled as clade *chr16*. For both subsets we ran independent simulations, starting 100 times from randomly generated DAGs, 100 times from an empty graph, and 100 times from the derived minimum spanning tree (dMST).

We generate random graphs by randomly picking a number k from the interval $(0, 3/2n)$, where n is the number of vertices, and then randomly choose k edges to add to an empty graph. If at a given step i, the insertion of the edge e_i introduces cycle, we discard this edge and pick again. As diverse real world networks are found to exibit some sort of scale-free properties [2] we also use the preferential attachment model to generate another set of random graphs.

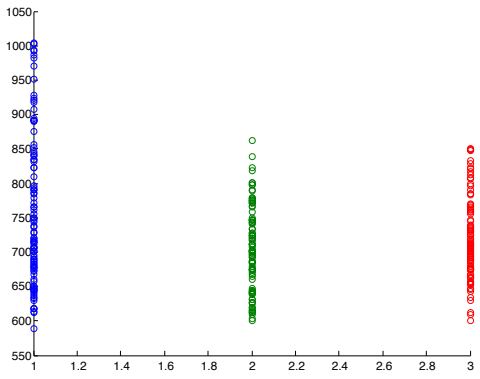

Figure 4: The optimal scores for chr10 achieved starting from (1) empty (2) mst (3) random DAGs

As we use directed acyclic graph to represent ancestral relationships (and moreover the duplication distance is asymetric $d(x, y) \neq d(y, x)$) the notion of minimum spanning tree might be vague. To clarify, we construct the dMST the following way. First, we find the MST for undirected graph U having the same vertices and weights on every edge the average of the two respective directed edges, i.e $w(e_{x,y}) = \frac{d(x,y)+d(y,x)}{2}$. This is similar to the use of MST in deriving relationships be-

tween retrotransposons [25, 24]. To obtain DAG, we impose directions on the edges of $MST(U)$ by looking into the minimum of $d(x, y)$ and $d(y, x)$).

We summarize the optimal scores for the subrgraph induced by clade *chr10* found during 100 independent trials from the three starting locations in Fig. 4. The three distribtions are quite similar having very close means and standard deviations: $\mu_{empty} = 692.31$, $\sigma_{empty} = 37.97$, $\mu_{mst} = 689.54$, $\sigma_{mst} = 35.79$, $\mu_{random} = 697.29$, $\sigma_{random} = 36.85$. Moreover, when we look at the the global optimum found for the three starting locations we observe that they are nearly identical ($opt_{empty} = 597$, $opt_{mst} = 601$, $opt_{random} = 601$). We obtained similar results by performing the same experiment on nodes belonging to clade *chr16*.

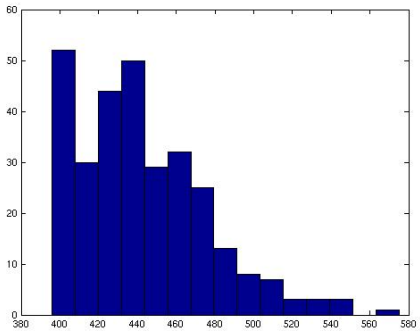

Figure 5: Results of 300 trials of simulated annealing (SA) heuristic: number of local optima returned by SA vs. objective scores. Results are from search for the max parsimony evolutionary history of clade 'chr16' from Fig. 3(a).

The simulated annealing heuristic often terminated in local optima. For a particular instance, the solutions found by all 300 trials would include many globally sub-optimal solutions. However, many of the locally optimal solutions encountered were "close" to the score for the best solution found. For example, the search for the max parsimony evolutionary history given in Fig. 3(a) resulted in a component whose objective score is 397; more than 1/6 of the total trials returned solutions whose objective scores are no more than 407 and well over 1/2 of the total trials returned solutions whose objective scores are no more than 437 (see Fig. 5).

4.5 Similiarity of Graphs with Near Optimal Scores

Another consideration is how similar the graphs that achieve near optimal scores are. We find that our algorithm can return fairly different DAGs even if these DAGs have identical scores. In fact, for clade *chr16* component, three DAGs with identical (and nearly optimal) parsimony scores have been found. To quantify more precisely these observations, we compute the frequency of each edge in the optimal DAGs for the entire set of 391 duplication blocks reached by simulated annealing. There are 536 edges (see Table 1) which appear in no more than 20% of the resulting graphs and these edges lead to fairly different topologies in the optimal DAGs. However, there are also 101 edges which are present in at least 90% of the solution. Moreover, certain small structures are often preserved. Considering again the example of the three different DAGs from clade *chr16* with identical scores, we find that the path $25 \rightarrow 276 \rightarrow 237 \rightarrow 241 \rightarrow 65$ is preserved in all of them.

num edges	frequency
55	100%
46	90%
69	80%
763	30-70%
281	20%
255	10%

Table 1: Edge frequency for the 300 optimal graphs.

4.6 Clades

Jiang et al. [13] performed an initial analysis of the duplication blocks in $\mathcal{D}$. They defined the distance between two duplication blocks as the Hamming distance between the binary vectors indicating the presence/absence of each duplicon. Note that this measure does not account for

the order, orientation, or multiplicities of duplicons in each block. They defined 24 *clades* of duplication blocks by performing hierarchical clustering on the Hamming distance matrix.

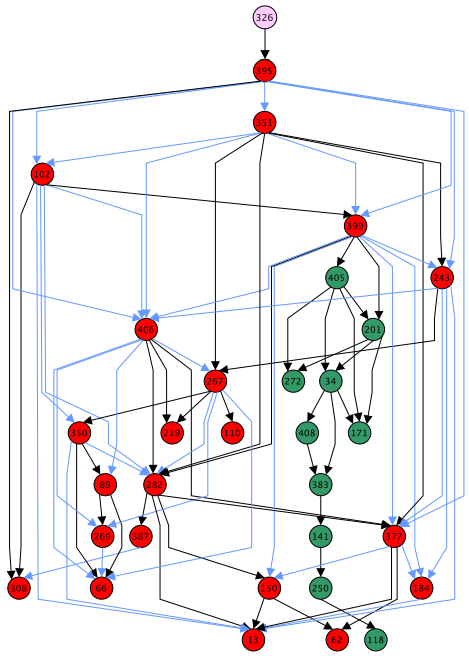

Figure 6: A connected component of the maximum parsimony ancestry DAG containing two clades: clade 'M1' is shown in red and clade 'chr7_2' is shown in green. Node labels correspond to duplication block IDs. The blue edges represents the inheritance network for non-core duplicon 6970.

We found a strong correspondence between clades and connected subgraphs in our duplication ancestry DAG. Four of the connected components with at least 4 nodes are comprised solely of duplication blocks belonging to the same clade. For example, Figure 2(a) shows the clade labeled 'chr10.' This clade corresponds to one connected component in our ancestry DAG plus four singleton nodes. Other connected components contain multiple clades, but the nodes of each clade often induce a connected subgraph indicating that the nodes within a clade are closely related. For example, the component in Figure 6 is comprised of clades 'M1' and 'chr7_2,' each of which induces a connected subgraph (with the exception of the two singleton nodes).

We quantify how "close" the duplication blocks in a clade are on our ancestry DAG, according to the following metric. For a clade C, let $\mathcal{D}_C$ be the set of nodes belonging to that clade and let G_C be the largest connected component in the subgraph of our ancestry DAG induced by $\mathcal{D}_C$. The relatedness metric is equal to $|G_C|/|\mathcal{D}_C|$. The mean over all 24 clades was 0.60, but the value of the relatedness metric varied considerably for different clades with clade 'M2' exhibiting the highest value (0.97) and clade 'chr7_4' exhibiting the lowest value (0.17) owing to a high number of singleton nodes in the clade.

The ancestry DAG not only defines groups of similar duplication blocks, but also shows direct ancestral relations within each clade, which was not possible using the hierarchical clustering of [13]. For example, for each clade, we can identify a seed block (or set of seed blocks) with no incoming edges in the subgraph induced by the clade. For example, the seed block in clade 'chr7_2' is 405 and the seed block in clade 'M1' is 395 (Fig. 6). We posit that these seed blocks appeared in the genome before the other members of their respective clades. Moreover, our analysis suggests that duplication events corresponding to copying substrings of seed blocks accounted for much of the subsequent construction of other members of their respective clades, indicating their importance in the expansion and fixation of duplication blocks within a clade.

Furthermore, the ancestry DAG indicates the relationships between different clades (Fig. 6)). For example, nodes in clade 'chr7_2' are all descended from node 399 in clade 'M1,' suggesting an ancestral relationship between these two clades. Similarly, clade 'M3_3' exhibits one seed block (190) that is it self a child of block 413 in clade 'M3_2.' Thus'M3_3' is descended from 'M3_2' in an optimal duplication scenario.

4.7 Core Duplicons

Jiang et al. [13] defined "core duplicons" as being any duplicon that appears in at least 67% of the duplication blocks of a given clade. They showed that core duplicons were enriched for genes and transcripts.

Implicit in the ancestry DAG is information about *inheritance networks* for each duplicon. For a given node $Y \in \mathcal{D}$, let $\mathcal{P}(Y)$ denote the set of parent nodes $\{X_i\}$ such that (X_i, Y) is an edge in our DAG. For every $(Y, \mathcal{P}(Y))$ pair, we can infer precisely which duplicons were copied from each respective parent in the optimal scenario to generate Y. Therefore, we can annotate the edges of our DAG with duplicons according to where each duplicon is passed from parent to child throughout the DAG. The inheritance network for a duplicon is the subgraph of the ancestry DAG induced on the edges on which that duplicon is passed from parent to child.

The inheritance networks for core duplicons corresponded to edges within subgraphs induced by a particular clade, as expected. Interestingly, we found duplicons that had not been identified by Jiang et al. as core duplicons but whose inheritance network included many edges within the subgraph induced by a particular clade. Most notably, duplicon 6970 appeared on 36 of the 63 total edges in the subgraph induced by clade 'M1' (shown in blue in Fig. 6) and does not appear on any other edge in the graph. By contrast, the maximum size of the inheritance network of a core duplicon was only 17. We propose 6970 as a new core duplicon for this clade and suggest that others like it should also be categorized as core duplicons.

4.8 Core Subsequences

A major advantage of computing duplication scenarios (and the associated feasible sets) in our duplication model is that we can compute the inheritance networks not only for single duplicons, but also for *substrings* or *subsequences* of duplicons. We found inheritance networks for many conserved subsequences that were nearly as prominent within particular clades as those for individual core duplicons. For example, the subsequence $[6968, 6967, 6925, 6963, 6962]$ of duplicons appears on 23 of the edges in the subgraph induced on 'M1' clade nodes We also indicate the inheritance networks for the subsequences $[7039, 7036, 7037]$ (shown in red) and $[9448, 9449]$ (shown in blue). The prevalence of conserved subsequences indicates that not only are individual duplicons important in the expansion and fixation of duplication blocks in a clade, but actually there are entire subsequences of duplicons that are frequently copied intact between nodes in a particular clade. This underscores the need to examine more than duplicon content when determining similarity between duplication blocks.

5 Discussion

Our maximum parsimony and maximum likelihood reconstructions show some significant differences, both from each other and from the analysis of Jiang et al. [13]. It will be interesting to see why certain maximum parsimony subgraphs have very low probability and what are the main reasons for the differences between the two reconstructions. Moreover, studying the newly identified core duplicons (not present in [13]) and core subsequences with respect to genes located nearby is highly promising.

From the perspective of analyzing our simulated annealing heuristic further examination of the effectiveness of the cooling schedule is warranted as is of the stability of the optimal graph. In addition, applying this optimization technique to other problems that could be modeled with DAGs has a great potential.

From the perspective of modeling segmental duplications, incorporating other types of operations, such as deletions and inversions, as well as single nucleotide mutations, would provide us with a more detailed picture. The phylogenetic problem where ancestral states (i.e. internal nodes and seed blocks in the DAG) are unknown remains open. Finally, it would be enlightening to compare the ancestral reconstructions of human segmental duplications to the segmental duplications in closely related primates.

6 Acknowledgments

The author would like to thank Ben Raphael and Crystal Kahn for their wonderful collaboration, Suzanne Sindi for the many insightful discussions, and Anna Ritz for her help.

References

[1] C. Alkan, J. M. Kidd, T. Marques-Bonet, G. Aksay, F. Antonacci, F. Hormozdiari, J. O. Kitzman, C. Baker, M. Malig, O. Mutlu, S. C. Sahinalp, R. A. Gibbs, and E. E. Eichler. Personalized copy number and segmental duplication maps using next-generation sequencing. *Nat. Genet.*, 41:1061–1067, 2009.

[2] L.A.N. Amaral, A. Scala, M. Barthelemy, and H.E. Stanley. Classes of small-world networks. *PNAS*, 97:11149–11152, 2000.

[3] Mir M. Atiqullah. An Efficient Simple Cooling Schedule for Simulated Annealing. In *Computational Science and Its Applications - ICCSA 2004*, pages 564–570. Springer Berlin, 2004.

[4] J.A. Bailey and E.E. Eichler. Primate Segmental Duplications: Crucibles of Evolution, Diversity and Disease. *Nat. Rev. Genet.*, 7:552–564, Jul 2006.

[5] J.A. Bailey and E.E. Eichler. Primate segmental duplications: crucibles of evolution, diversity and disease. *Nat. Rev. Genet.*, 7:552–564, 2006.

[6] R. Blekhman, A. Oshlack, and Y. Gilad. Segmental duplications contribute to gene expression differences between humans and chimpanzees. *Genetics*, 182:627–630, 2009.

[7] David Maxwell Chickering. Learning bayesian networks is np-complete. In *Learning from Data: Artificial Intelligence and Statistics V*, pages 121–130. Springer-Verlag, 1996.

[8] F.D. Ciccarelli, C. von Mering, M. Suyama, E.D. Harrington, E. Izaurralde, and P. Bork. Complex genomic rearrangements lead to novel primate gene function. *Genome Res.*, 15:343–351, 2005.

[9] G. Dueck and T. Scheuer. Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing. *J. Comp. Phys.*, 90:161–175, 1990.

[10] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell., PAMI-6*, 6:721–741, 1984.

[11] Paolo Giudici and Robert Castelo. Improving markov chain monte carlo model search for data mining. *Machine Learning*, 50(1-2):127–158, 2003.

[12] Bruce Hajek. Cooling schedules for optimal annealing. *Math. Oper. Res.*, 31:311–329, 1988.

[13] Zhaoshi Jiang, Haixu Tang, Mario Ventura, Maria Francesca Cardone, Tomas Marques-Bonet, Xinwei She, Pavel A Pevzner, and Evan E Eichler. Ancestral reconstruction of segmental duplications reveals punctuated cores of human genome evolution. *Nature Genetics*, 39:1361–1368, 2007.

[14] C.L. Kahn, B.H. Hristov, and B.J. Raphael. Parsimony and Likelihood Reconstruction of Human Segmental Duplications. *in submission*.

[15] Crystal L. Kahn, Shay Mozes, and Ben J. Raphael. Efficient Algorithms for Analyzing Segmental Duplications, Deletions, and Inversions in Genomes. In *Lecture Notes in Computer Science*, volume 5724/2009, pages 169–180, 2009.

[16] Crystal L. Kahn and Ben J. Raphael. Analysis of Segmental Duplications via Duplication Distance. *Bioinformatics*, 24:i133–138, 2008.

[17] Crystal L. Kahn and Ben J. Raphael. A Parsimony Approach to Analysis of Human Segmental Duplications. In *Pacific Symposium on Biocomputing*, pages 126–137, 2009.

[18] P. M. Kim, H. Y. Lam, A. E. Urban, J. O. Korbel, J. Affourtit, F. Grubert, X. Chen, S. Weissman, M. Snyder, and M. B. Gerstein. Analysis of copy number variants and segmental duplications in the human genome: Evidence for a change in the process of formation in recent evolutionary history. *Genome Res.*, 18:1865–1874, 2008.

[19] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220:671–680, 1983.

[20] E.V. Linardopoulou, S.S. Parghi, C. Friedman, G.E. Osborn, S.M. Parkhurst, and B.J. Trask. Human subtelomeric WASH genes encode a new subclass of the WASP family. *PLoS Genet.*, 3:e237, 2007.

[21] J.R. Lupski and P. Stankiewicz. Genomic disorders: molecular mechanisms for rearrangements and conveyed phenotypes. *PLoS Genet.*, 1:e49, 2005.

[22] N. Metropolis, A.W. Rosenbluth, M. N. Rosenbluth, A.H. Teller, and E. Teller. Equations of State Calculations by Fast Computing Machines. *J. Chem*, 21:1087–1092, 1958.

[23] Yaghout Nourani and Bjarne Andresen. A comparison of simulated annealing cooling strategies. *J. Phys. A: Math. Gen.*, 31:8373–8385, 1998.

[24] Sean O'Rourke, Noah Zaitlen, Nebojsa Jojic, and Eleazar Eskin. Reconstructing the Phylogeny of Mobile Elements. In *In Proceedings of the Eleventh Annual Conference on Research in Computational Biology (RECOMB 2007)*, pages 196–210, Berlin, 2007. Springer.

[25] A.L. Price, E. Eskin, and P.A. Pevzner. Whole-genome analysis of Alu repeat elements reveals complex evolutionary history. *Genome Res.*, 14:2245–2252, 2004.

[26] B.J. Raphael and P.A. Pevzner. Reconstructing tumor amplisomes. *Bioinformatics*, 20 Suppl 1:i265–273, 2004.

[27] K. Vandepoele, N. Van Roy, K. Staes, F. Speleman, and F. van Roy. A novel gene family NBPF: intricate structure generated by gene duplications during primate evolution. *Mol. Biol. Evol.*, 22:2265–2274, 2005.

[28] M. C. Zody, Z. Jiang, H. C. Fung, F. Antonacci, L. W. Hillier, M. F. Cardone, T. A. Graves, J. M. Kidd, Z. Cheng, A. Abouelleil, L. Chen, J. Wallis, J. Glasscock, R. K. Wilson, A. D. Reily, J. Duckworth, M. Ventura, J. Hardy, W. C. Warren, and E. E. Eichler. Evolutionary toggling of the MAPT 17q21.31 inversion region. *Nat. Genet.*, 40:1076–1083, 2008.

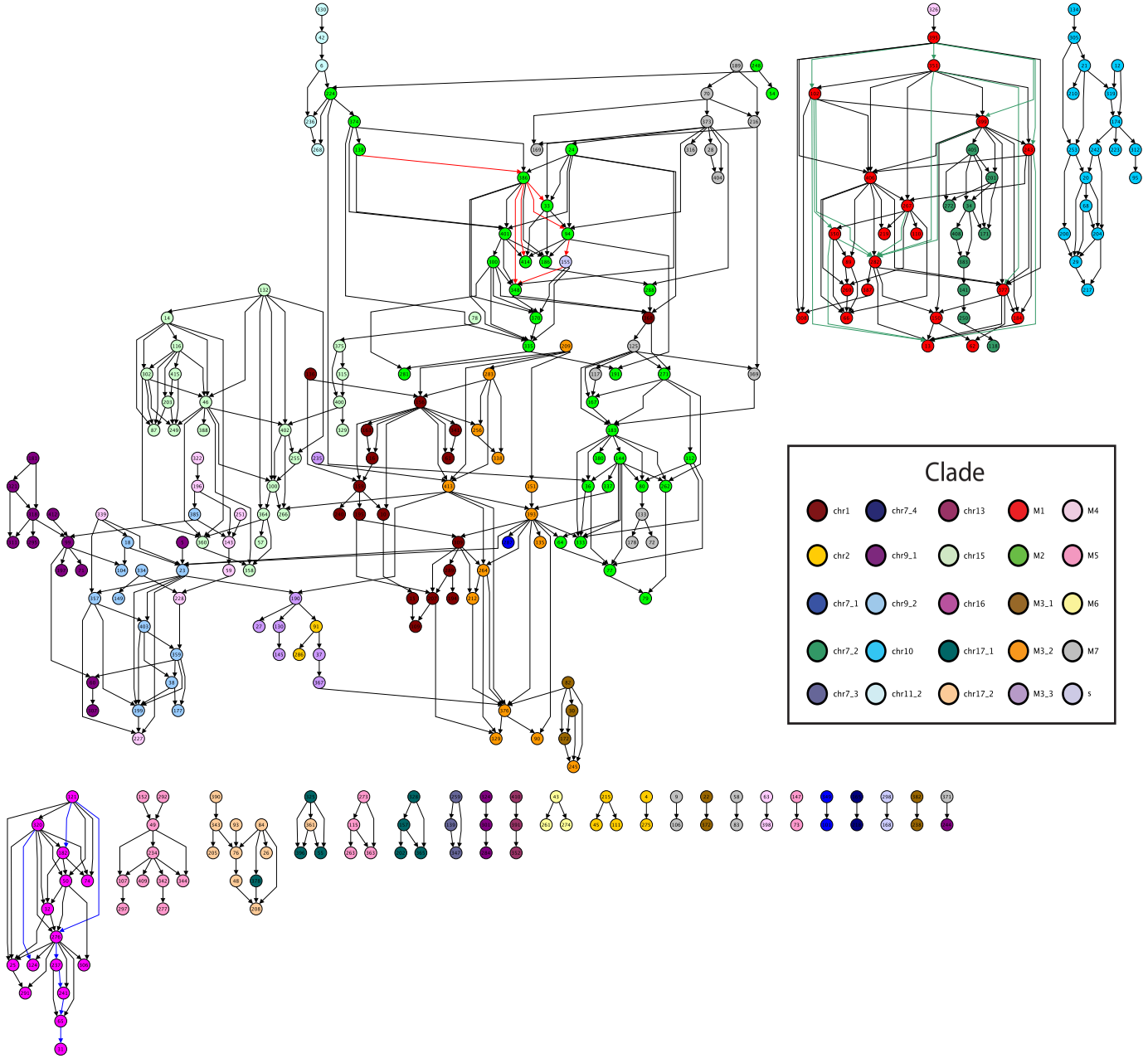

Figure 7: The maximum parsimony DAG for a set of 391 duplication blocks in the human genome. The nodes represent duplication blocks. Edges indicate evolutionary relations; an edge is directed from a node u to a node v if the most-parsimonious duplication scenario includes duplication events that copy substrings of u in the construction of v. Jiang et al. NG partitioned the duplication blocks into a set of 24 clades (plus one 's' group of duplication blocks found in subtelomeric regions) that we indicate here with 25 colors on nodes. The 3 sets of colored edges represent inheritance networks for 3 conserved subsequences of duplicons. These inheritance networks are almost entirely confined to a single clade each. The green edges represent the inheritance of the duplicon sequence $[6968, 6967, 6965, 6963, 6962, 6960]$ in clade 'M1', the red edges represent the inheritance of $[7039, 7036, 7037]$ in clade 'M2', and the blue edges represent the inheritance of $[9448, 9449]$ in clade 'chr16.'