BIOINFORMATICS

Vol. 26 ECCB 2010, pages i446–i452
doi:10.1093/bioinformatics/btq368

Parsimony and likelihood reconstruction of human segmental duplications

Crystal L. Kahn[1,*], Borislav H. Hristov[1] and Benjamin J. Raphael[1,2,*]

[1]Department of Computer Science and [2]Center for Computational Molecular Biology, Brown University, Providence, RI, 02912, USA

ABSTRACT

Motivation: Segmental duplications >1 kb in length with $\geq 90\%$ sequence identity between copies comprise nearly 5% of the human genome. They are frequently found in large, contiguous regions known as *duplication blocks* that can contain mosaic patterns of thousands of segmental duplications. Reconstructing the evolutionary history of these complex genomic regions is a non-trivial, but important task.

Results: We introduce parsimony and likelihood techniques to analyze the evolutionary relationships between duplication blocks. Both techniques rely on a generic model of duplication in which long, contiguous substrings are copied and reinserted over large physical distances, allowing for a duplication block to be constructed by aggregating substrings of other blocks. For the likelihood method, we give an efficient dynamic programming algorithm to compute the weighted ensemble of all duplication scenarios that account for the construction of a duplication block. Using this ensemble, we derive the probabilities of various duplication scenarios. We formalize the task of reconstructing the evolutionary history of segmental duplications as an optimization problem on the space of directed acyclic graphs. We use a simulated annealing heuristic to solve the problem for a set of segmental duplications in the human genome in both parsimony and likelihood settings.

Availability: Supplementary information is available at http://www.cs.brown.edu/people/braphael/supplements/.

Contact: clkahn@cs.brown.edu; braphael@cs.brown.edu.

1 INTRODUCTION

A striking feature of mammalian genomes is the prevalence of segmental duplications or low-copy repeats. Approximately 5% of the human genome consists of segmental duplications >1 kb in length with ≥90% sequence identity between copies (Bailey and Eichler, 2006). 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 (Blekhman *et al.*, 2009).

Segmental duplications remain an extreme challenge for evolutionary reconstruction, as they are the 'most structurally complex and dynamic regions of the human genome' (Alkan *et al.*, 2009). Human segmental duplications are frequently found within complicated mosaics of duplicated fragments (Bailey and Eichler, 2006). Jiang *et al.* (2007) 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.

Numerous authors have considered the problem of analyzing relationships between genome sequences that contain duplicated segments. This work falls into roughly two categories. The first focus is the problem of computing genome rearrangement distances, like reversal distance, in the presence of duplicated genes or synteny blocks (El-Mabrouk, 2002; Marron *et al.*, 2004; Sankoff, 1999, for example). However, such rearrangement distances do not model the creation of new duplicates and thus are not well-suited to describe the evolutionary history of segmental duplications in the genome. The second focus is to analyze regions with duplications under 'local' operations like tandem duplications (Chaudhuri *et al.*, 2006; Lajoie *et al.*, 2007, for example). While tandem duplication is undoubtedly important in the generation of duplication blocks, there is strong evidence that an important characteristic of the history of segmental duplications is the frequent duplication and transposition of long segments over large physical distances; as many as 50–60% of segmental duplications were transposed interchromosomally (Bailey and Eichler, 2006). Several general models of rearrangement that allowed for both local operations and duplication–transposition-like operations between different strings were studied by Ergun *et al.* (2003), but the generality of those models meant that the distances were NP-hard to compute and only approximation algorithms were given.

Here, we present a novel formulation of the problem of computing an evolutionary history for a set of segmental duplications that are organized in duplication blocks. We represent evolutionary relationships between a set of duplication blocks as a directed acyclic graph (DAG), and we formalize the evolutionary reconstruction problem as an optimization over the space of DAGs.

We present two different methods for scoring a DAG: one based on parsimony and one based on likelihood. The parsimony score for a DAG is a straightforward extension of 'duplication distance', a measure introduced by some of us (Kahn and Raphael, 2008, 2009) that describes the most parsimonious sequence of duplicate operations needed to construct a given target string. The likelihood score for a DAG is the product of the likelihood scores for each of the duplication blocks, where a duplication block's likelihood is derived by computing the weighted ensemble of all possible duplication scenarios that could have generated it. We describe how to compute the partition function of the ensemble efficiently using a dynamic program that generalizes the duplication distance (i.e. parsimony score) recurrence. Deriving a probabilistic model from a dynamic program this way is analogous to the

*To whom correspondence should be addressed.

approach of McCaskill (1990) who applied dynamic programming to RNA folding to compute the partition function of all secondary structures and to assign probabilities to certain substructures..

Finally, we solve these evolutionary reconstruction problems on the set of duplication blocks identified by Jiang *et al.* (2007) using a local search technique based on simulated annealing. We compare these reconstructions to the analysis of Jiang *et al.* (2007). Our evolutionary reconstruction recapitulates some of the properties of earlier analysis but also reveals additional and more subtle relationships between segmental duplications.

2 METHODS

Here, we present two methods for determining the optimality of an evolutionary relationship between a pair of duplication blocks—one based on a parsimony criterion and one based on a likelihood criterion. In Sections 2.1 and 2.2, we describe the parsimony-based model of segmental duplication that is based on *duplication distance*, introduced in Kahn and Raphael (2008, 2009). Next, 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.

2.1 A model of segmental duplication

As noted above, an important characteristic of segmental duplications that distinguishes them from other types of repeats is that they are frequently transposed across large genomic distances from their respective ancestral loci. In Kahn and Raphael (2008, 2009), we modeled the process in which a duplication block, a composite of many duplicons, is built by copying strings of duplicons from *other* duplication blocks. In particular, we define the basic ‘copy–paste’ operation as follows.

Definition 2.1. *A* duplicate operation, $\delta_{s,t,p}(X)$, *copies a substring* $X_{s,t}$ *of a source string* X *and pastes it into a target string at position* p.[1] *Specifically, if* $X = x_1 \ldots x_m$ *and* $Z = z_1 \ldots z_n$, *then* $Z \circ \delta_{s,t,p}(X) = z_1 \ldots z_{p-1} x_s \ldots x_t z_p \ldots z_n$.

Definition 2.2. *The* duplication distance,[2] $d(X,Y)$, *from a source string* X *to a target string* Y *is the minimum number of duplicate operations needed to construct* Y *by copying and pasting substrings of* X *into an initially empty target string.*

A *subsequence* is distinguished from a substring because the characters of a subsequence need not be contiguous. Given a string X, a subsequence S of X can be expressed as an increasing list of indices of X. For example, for $X = abcdefg$, the subsequence $S = (1,3,5)$ is the string *ace*.

Definition 2.3. *Two subsequences* $S = (s_1, s_2, \ldots, s_{l_s})$ *and* $T = (t_1, t_2, \ldots, t_{l_t})$ *of a string* X overlap *if either (i) there exist indices* $i: 1 \leq i < l_s$ *and* $j: 1 \leq j < l_t$ *such that* $i = j$, *or (ii) there exist indices* $i, i': 1 \leq i < i' < l_s$ *and a* $j, j': 1 \leq j < j' < l_t$ *such that either* $i < j < i' < j'$ *or* $j < i < j' < i'$.

Given a source/target pair X, Y, any sequence of duplicate operations of the form $\delta_{s_1,t_1,p_1}(X), \ldots, \delta_{s_d,t_d,p_d}(X)$ that generates Y from X uniquely partitions the characters of Y into non-overlapping subsequences corresponding to characters that were copied conjointly from X.

Definition 2.4. *Given a source string* X, *a* generator $\Psi_X = (X_{i_1,j_1}, \ldots, X_{i_k,j_k})$ *is a sequence of substrings of* X.

$$\begin{aligned} X &= abcde \\ Y_0 &= \emptyset \\ Y_1 &= Y_0 \circ \delta_{1,3,1}(X) = abc \\ Y_2 &= Y_1 \circ \delta_{4,5,1}(X) = deabc \\ Y &= Y_2 \circ \delta_{4,5,5}(X) = deabdec \end{aligned}$$

Fig. 1. An example of a sequence of duplicate operations that constructs $Y = deabdec$ from $X = abcde$. The corresponding feasible generator is: $\Psi_X = (X_{4,5}, X_{1,3}, X_{4,5}) = ((de),(abc),(de))$.

Definition 2.5. *A generator* $\Psi_X = (X_{i_1,j_1}, \ldots, X_{i_k,j_k})$ *is* feasible *for a target string* Y, *that we denote as* $\Psi_X \dashv Y$, *if:*

1. *The elements of* Ψ_X *partition the characters of* Y *into mutually non-overlapping subsequences* $\{S_1, \ldots, S_k\}$.
2. *There exists a bijective mapping* $f: \{X_{i,j} \in \Psi_X\} \to \{S_1, \ldots, S_k\}$ *from substrings of* X *to subsequences in* Y *corresponding to how the elements of* Ψ_X *partition* Y.
3. *The order of elements in* Ψ_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.

See Figure 1.

A sequence of k duplicate operations that constructs Y from X uniquely defines a feasible generator Ψ_X with length k whose elements correspond, respectively, to substrings of X that are duplicated conjointly in a single operation.

2.2 Parsimony

In Kahn *et al.* (2010), we describe a polynomial-time algorithm to compute the duplication distance from X to Y. We use duplication distance to measure the similarity between a pair of duplication blocks by counting the number of operations needed to generate Y from X in a simplest or most-parsimonious scenario.

While the parsimony assumption 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. Consider, for example, the strings $X =$ ‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, and ‘g’, hijkl, and $Y = agdbhecifdajebkfclg$. 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, in the next section, we present a probabilistic model of segmental duplication that considers the weighted ensemble of all feasible generators for a source/target string pair.

2.3 The partition function

For a given source string X and positive integer k, we consider the space of all length-k generators Ψ_X. We define a probability distribution on the collection of generators by defining $Pr[\Psi_X] \propto \omega(\Psi_X)$ where $\omega(\Psi_X)$ is the ‘score’, or weight, assigned to a generator, and we compute the partition function $Z_X^{(k)}$ of the weighted ensemble of all possible length-k generators Ψ_X. Given a source string X and a target string Y, we define the event F to be the event of choosing a length-k generator that is feasible for Y from the space of length-k generators. We define a probabilistic model for segmental duplications that, given a target string Y, assigns a probability to F: $Pr[F|Y,X,k]$. For a *fixed target string* Y, the probability, $Pr[F|Y,X,k]$, is the weighted ensemble of all possible length-k generators that are feasible for Y, normalized by the

[1] In (Kahn and Raphael, 2008, 2009), we also considered duplicate reversals in which the copied substring is inverted before being inserted into the target. We note that all of our definitions and algorithms presented here can be similarly augmented but we omit the details.

[2] We note that the duplication distance between a pair of strings is not formally a distance as it is asymmetric.

partition function $Z_X^{(k)}$. In particular, we can express the probability as:

$$Pr[F|Y,X,k] = \frac{1}{Z_X^{(k)}} \sum_{\Psi_X \dashv Y : |\Psi_X| = k} \omega(\Psi_X), \qquad (1)$$

where $|\Psi_X|$ denotes the length of the generator. The likelihood of a target string Y then can be expressed as $L(Y|F,X,k) = Pr(F|Y,X,k)$.

The score of a generator, $\omega(\Psi_X)$, can be defined according to various biological models. Although different functions ω may require different algorithms for computing the value $Pr[F|Y,X,k]$, we found that functions of the form $\omega(\Psi_X) = \sigma(|\Psi_X|, l(\Psi_X))$ where $l(\Psi_X) = \sum_{X_{i,j} \in \Psi_X} |X_{i,j}|$ denotes the sum of the lengths of the elements of Ψ_X, admit particularly efficient algorithms for computing Equation (1). We discuss the score function further in Supplementary Section 1.2.

Now, we give an algorithm to compute the partition function, $Z_X^{(k)}$. Given a score function of the form $\sigma(|\Psi_X|, l(\Psi_X))$, each length-$k$ generator whose elements have lengths that sum to l are scored the same, namely $\sigma(k,l)$. Therefore, in order to compute $Z_X^{(k)}$, we must calculate the total number of length-k generators whose lengths sum to l for all relevant values of l. Let $\mathcal{C}_X^{(k)}(l)$ equal the number of distinct length-k generators for which the sum of the lengths of the elements equals l.[3]

Lemma 2.6. *Let $X = x_1 \ldots x_{|X|}$ be a source string and let k and l be positive integers. The function $\mathcal{C}_X^{(k)}(l)$ satisfies the following recurrence.*

$$\mathcal{C}_X^{(1)}(l) = |X| - l + 1,$$

$$\mathcal{C}_X^{(k)}(l) = \sum_{l'=l-|X|}^{l-1} \mathcal{C}_X^{(k-1)}(l') \cdot (|X| - (l - l') + 1).$$

For a source string X and integers k, l, if we are given $\mathcal{C}_X^{(k)}(l)$, we can compute $Z_X^{(k)}$ efficiently by summing $\mathcal{C}_X^{(k)}(l)$ over all relevant lengths l, weighting each feasible generator appropriately according to the function $\sigma(k,l)$.

Theorem 2.7. *Let $X = x_1 \ldots x_{|X|}$ be a source string and k be a positive integer. The partition function $Z_X^{(k)}$ satisfies the following.*

$$Z_X^{(k)} = \sum_{l=k}^{|X| \cdot k} \mathcal{C}_X^{(k)}(l) \cdot \sigma(k,l).$$

Note that the elements of a length-k list of substrings of X can have lengths that sum to at least k and at most $|X| \cdot k$.

The recurrence in Lemma 2.6 can be computed in $O(|X|k)$ time, so $Z_X^{(k)}$ can be computed in $O(|X|^2 k^2)$ time according to Theorem 2.7. We omit a proof of correctness due to space considerations.

2.4 Restricted partition function

In this section, we present the final ingredient necessary to compute the probability $Pr[F|Y,X,k]$, namely the sum in Equation (1) that we define as $Q_X^{(k)}(Y)$. We refer to the value $Q_X^{(k)}(Y)$ as the *restricted partition function of feasible generators*, and it is equal to the weighted ensemble of all length-k generators Ψ_X that are feasible for Y. Hence $Q_X^{(k)}(Y) = \sum_{\Psi_X \dashv Y : |\Psi_X| = k} \omega(\Psi_X) = \sum_{\Psi_X \dashv Y : |\Psi_X| = k} \sigma(k, |Y|)$.

In order to compute this value, we generalize the recurrence presented in Kahn *et al.* (2010) for computing duplication distance from source string X to target string Y to count the number of length-k generators that are feasible for Y.

Lemma 2.8. *Given a source string $X = x_1 \ldots x_{|X|}$ and a target string $Y = y_1, \ldots, y_{|Y|}$, the number $N_X^{(k)}(Y)$ of distinct length-k generators Ψ_X that are feasible for Y satisfies the following recurrence.*

$$N_X^{(k)}(Y) = \sum_{i : x_i = y_1} N_X^{(k)}(Y,i),$$

$$N_X^{(1)}(Y,i) = \begin{cases} 1 & \textit{if } Y = X_{i,i+|Y|-1}, \\ 0 & \textit{otherwise,} \end{cases}$$

$$N_X^{(k)}(Y,i) = N_X^{(k-1)}(Y_{2,|Y|}) + \sum_{j>1 : y_j = x_{i+1}} \sum_{l=1}^{k} [N_X^{(l)}(Y_{2,j-1}) \cdot N_X^{(k-l)}(Y_{j,|Y|}, i+1)].$$

Here, the term $N_X^{(k)}(Y,i)$ represents the number of feasible generators Ψ_X with length k given that the character y_1 is generated by a substring of X starting at x_i.

We compute the restricted partition function $Q_X^{(k)}(Y)$ efficiently by first counting the number of relevant feasible generators, namely $N_X^{(k)}(Y)$, and scoring each generator appropriately by $\sigma(k, |Y|)$.

Theorem 2.9. *Let $X = x_1 \ldots x_{|X|}$, $Y = y_1, \ldots, y_{|Y|}$ be a source/target string pair and let k be a positive integer. The restricted partition function $Q_X^{(k)}(Y)$ satisfies the following.*

$$Q_X^{(k)}(Y) = N_X^{(k)}(Y) \cdot \sigma(k, |Y|).$$

The recurrence given in Lemma 2.8 can be computed in time $O(|Y|^2 k^2 \mu(Y) \mu(X))$ where $\mu(Y)$ (resp. $\mu(X)$) is the maximum multiplicity of any character that appears in Y (resp. X), so computing $Q_X^{(k)}(Y)$ takes the same time. We include a proof of correctness in Supplementary Section 1.1.

3 ALGORITHM

Here, 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.

3.1 Maximum parsimony and maximum likelihood evolutionary histories

The input to our 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. We assume that the ancestral genome 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 constructs the remaining duplication blocks by duplicating substrings of previously constructed blocks.

We observed in Kahn and Raphael (2008) strong evidence that many of the duplication blocks identified by Jiang *et al.* (2007) had been constructed through the duplication and aggregation of substrings of duplicons from several other blocks. Therefore, a tree cannot aptly represent an evolutionary history; a more appropriate representation of the evolutionary relationships between duplication blocks is a DAG in which the vertices represent duplication blocks and an edge directed from a vertex X to a vertex Y indicates that

[3] The value $\mathcal{C}_X^{(k)}(l)$ is related to the well-known integer partition function $p(n)$ and corresponding Young tableaux. If $\mathcal{P}(l,k)$ is the set of partitions of the integer l into k parts, we can express $\mathcal{C}_X^{(k)}(l) = \sum_{P \in \mathcal{P}(l,k)} \sum_{\mathrm{p} \in P} (|X| - p + 1) \cdot k!$.

substrings of X were duplicated in the construction of Y. A vertex with multiple incoming edges and, therefore, multiple parents, is constructed using substrings of all of the parent blocks. Specifically, given a DAG $G=(\mathcal{D},E)$, for $Y\in\mathcal{D}$, we define $P_G(Y)$, the parent string of Y, by $P_G(Y)=X_1\odot X_2\odot\cdots\odot X_p$ where $X_i\in\{\mathcal{D}|(X_i,Y)\in E\}$ and $\odot$ indicates the concatenation of two strings with a dummy character inserted in between.

We make two 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, ensuring the evolutionary 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. Similar assumptions were made, for example, by Price *et al.* (2004) to derive the evolutionary tree for Alu repeat elements.

We can define the optimal DAG with respect to a parsimony criterion using duplication distance (Definition 2.2).

DEFINITION 3.1. *Given a set of duplication blocks $\mathcal{D}$, the* maximum parsimony evolutionary history *is the DAG $G=(\mathcal{D},E)$ that minimizes $f(G)=\sum_{Y\in\mathcal{D}}d(P_G(Y),Y)$.*

We can also define the optimal DAG with respect to a likelihood criterion. In phylogenetic tree reconstruction, a max 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 Section 2. We maximize the likelihood of the solution over all DAG topologies and—instead of branch lengths—the numbers of operations permitted to construct each node.

DEFINITION 3.2. *Given a set of duplication blocks $\mathcal{D}$, the* maximum likelihood evolutionary history *is the DAG $G=(\mathcal{D},E)$ that maximizes the likelihood:*

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

where $Z^{(k)}_{P_G(X)}$ and $Q^{(k)}_{P_G(Y)}$ are the partition function and restricted partition functions, respectively.

4 IMPLEMENTATION

We analyzed a set of 391 duplication blocks identified by Jiang *et al.* (2007) that were represented as signed strings on an alphabet of ≈11 000 duplicons. We computed the maximum parsimony evolutionary history (Definition 3.1) for the entire set of blocks (Fig. 2). The DAG exhibited multiple connected components. For comparison, we then computed the maximum likelihood evolutionary histories (Definition 3.2) for several of the subgraphs induced by connected components of the parsimony solution. We scored generators according to $\sigma(k,|Y|)=\frac{1}{|Y|^k}$ (see Supplementary Section 1.2).

We used a simulated annealing strategy to find a maximum parsimony DAG for the entire set of duplication blocks and to find maximum likelihood DAGs for several subgraphs[4] (see Supplementary Section 1.3 for details). For each input, we ran our local search 300 times. We started the search an equal number of times at each of three different types of initial graphs: (i) the empty graph with no edges; (ii) the directed minimum spanning tree (MST); and (iii) a randomly chosen DAG (chosen independently for each trial). Finally, to focus the search on the most important block relationships, we considered only edges between blocks whose longest common subsequence (LCS) contained at least 20 duplicons.

4.1 Maximum parsimony reconstruction

The maximum parsimony DAG contains 391 nodes and 479 edges. There are nine connected components with at least four duplication blocks, and nearly 40% of the blocks appear in the largest connected component. Figure 3 shows a moderately-sized connected component. The graph also contains a total of 105 singleton nodes for which we did not infer any evolutionary relations with other duplication blocks, 97 of which did not exhibit an LCS of length 20 with any other block.

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.

4.2 Clades and core duplicons

Jiang *et al.* (2007) performed an initial analysis of the duplication blocks. They defined 24 *clades*, or groups of duplication blocks derived from a common ancestor block, by performing hierarchical clustering on a matrix representing the shared presence or absence of duplicons for every pair of blocks. For a given clade they defined a *core duplicon* as one that appears in at least 67% of the constituent duplication blocks. They posited that clades represent families of evolutionarily related duplication blocks and that core duplicons ‘may have driven the evolution of the duplication blocks’ in a clade.

After construction, we colored the nodes of our DAG according to the clades described in Jiang *et al.* (2007). We found a strong correspondence between Jiang *et al.*’s clades and connected subgraphs in our DAG; five of the nine connected components with at least four blocks were comprised of duplication blocks belonging to a single clade and seven of the nine components were comprised of blocks belonging to no more than two clades. For example, see Figures 4a and 5a. In larger components, nodes from a single clade frequently induce a connected subgraph. For example, see Figure 3.

Our DAG also reveals which duplication blocks may have seeded many other blocks (i.e. those with high out-degree). For example, in Figure 3, block 399 exhibits eight children and is an inflection point for the component. Moreover, the edge from block 399 to 405

[4] Both the max parsimony and max likelihood versions of the problem can be shown to be NP-hard by a reduction from the problem of Learning Bayesian Networks.

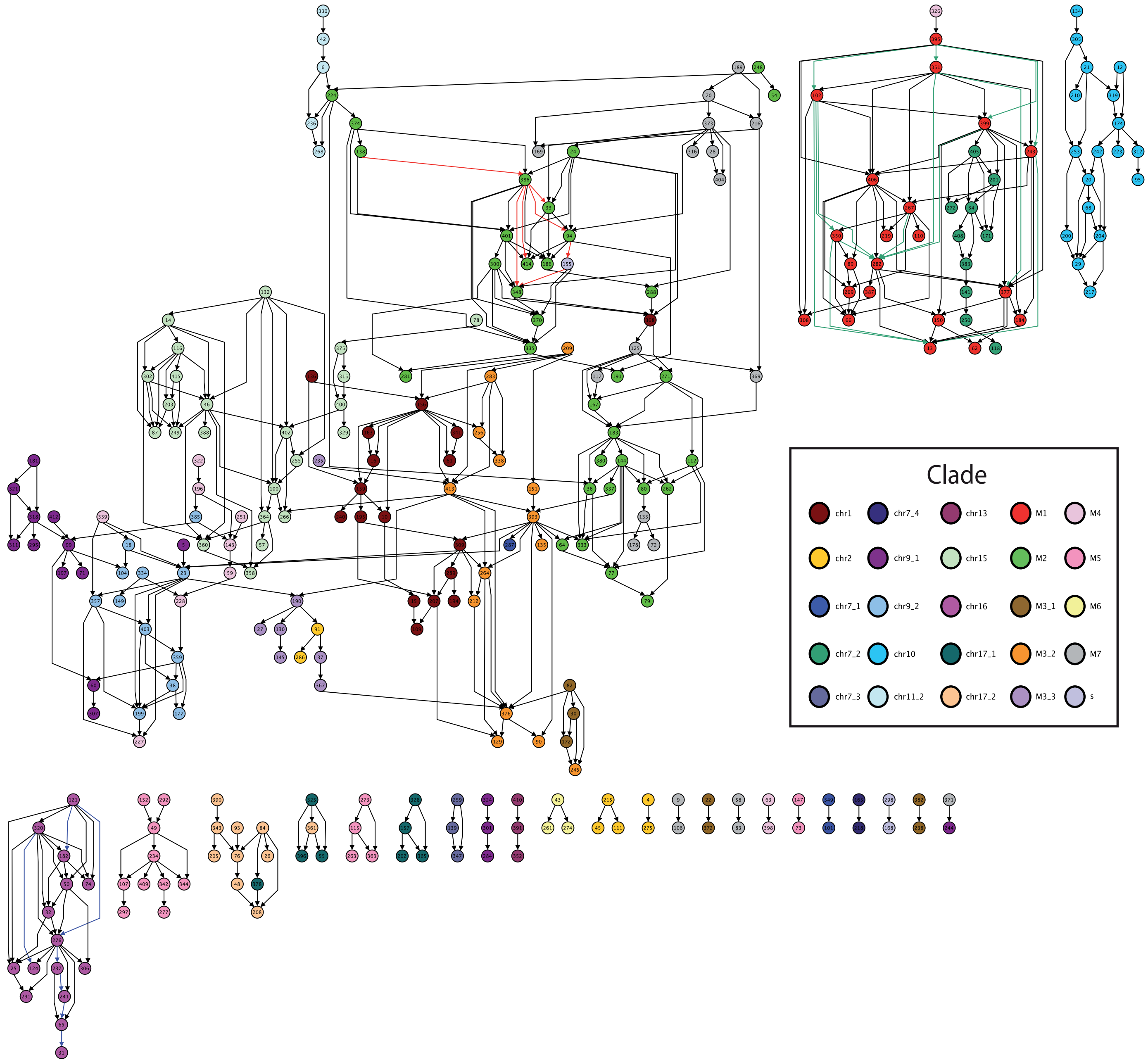

Fig. 2. The maximum parsimony DAG for a set of 391 duplication blocks in the human genome. The nodes represent duplication blocks. Edges indicate evolutionary relations; an edge is directed from a node *u* to a node *v* if the most-parsimonious duplication scenario includes duplication events that copy substrings of *u* in the construction of *v*. Jiang *et al.* (2007) 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.'

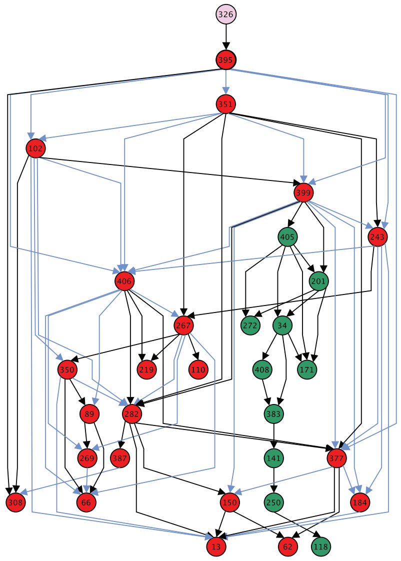

Fig. 3. A connected component of the maximum parsimony DAG. Nodes from clade 'M1' are red and nodes from clade 'chr7_2' are green. Node labels correspond to duplication block IDs. The blue edges represents the inheritance network for non-core duplicon 6970.

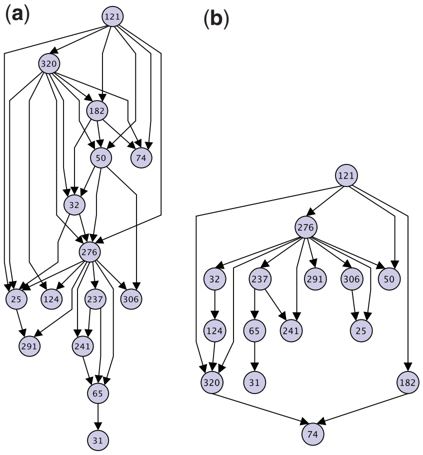

Fig. 4. (**a**) Component comprised entirely of duplication blocks from clade 'chr16' in the maximum parsimony DAG. (**b**) Maximum likelihood DAG for subgraph induced on nodes in (a).

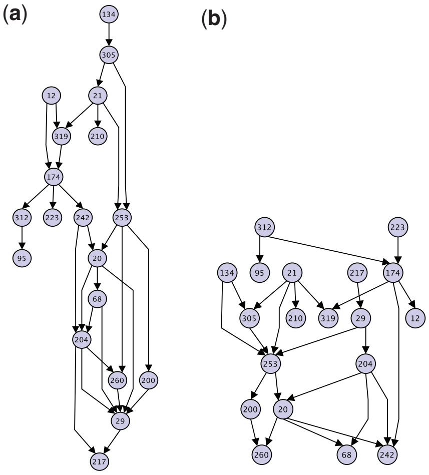

Fig. 5. (**a**) Component comprised entirely of duplication blocks from clade 'chr10' in the maximum parsimony DAG. (**b**) Maximum likelihood DAG for subgraph induced on nodes in (a).

links blocks from the the 'M1' and 'chr7_2' clades. Even though the blocks 399 and 405 belong to different clades, 405 is very 'close' to 399 in duplication distance: block 405 contains only 71 duplicons, but it shares a subsequence of 29 duplicons with block 399. This link suggests that the entirety of clade 'chr7_2' was descended from clade 'M1' in an optimal duplication history.

Also implicit in the DAG is information about which duplicons are duplicated from one block to another in an optimal duplication history. We define the *inheritance network* for each duplicon as the subgraph induced on the edges on which that duplicon is passed from parent to child. Interestingly, a comparison of the inheritance networks for core and non-core duplicons revealed that many non-core duplicons exhibit larger inheritance networks within subgraphs induced by a clade than core duplicons. For example, non-core duplicon 6970 appeared on 36 of the 63 total edges in the subgraph induced by clade 'M1' (shown in blue in Fig. 3) and does not appear on any other edge in the graph. In 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.

Moreover, we found inheritance networks for many conserved subsequences of duplicons that were nearly as prominent 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 by 'M1' clade nodes (shown as green edges in Fig. 2). Similarly, the sequence [7039, 7036, 7037] exhibits a connected inheritance network of 7 edges within the subgraph induced on clade 'M2', and [9448, 9449] exhibits an inheritance network of seven edges within the subgraph induced on clade 'chr16' that includes an inheritance path of length 5 (Fig. 2). By delineating the inheritance networks of duplicon subsequences that are conserved across duplication blocks, we can learn about which duplicons were duplicated and transposed conjointly. This type of analysis was impossible using only the clade annotations of Jiang *et al.* (2007).

4.3 Maximum likelihood reconstruction

We computed the maximum likelihood DAGs (Definition 3.2) 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 (Fig. 2). The maximum likelihood subgraphs for these subproblems are shown in Figures 4b and 5b.

The two DAGs for the 'chr16' component in Figure 4 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. Both solutions are similarly 'good' with respect to the parsimony objective: the solution in (a) exhibits an optimal parsimony score of 397, and the one in (b) exhibits a score of 401.

However, the likelihood score for the parsimony solution in (a) was nearly zero. One difference that accounts for this discrepancy is the higher average in-degree for blocks in the parsimony solution (2.2) as compared to the likelihood solution (1.3). Also, the parsimony solution exhibits a path with ten edges, whereas the longest path in the likelihood solution has six.

Some of these differences are due to the fact that the parsimony criterion does not penalize edges that do not directly improve the score. For example, block 291 has two parents (276 and 25) in the parsimony DAG but only one parent (276) in the likelihood DAG. However, the duplication distance with source 276 $\odot$ 25 and target 291 is the same as the duplication distance with source 276 and target 291. Therefore, the edge from 25 to 291 does not improve the parsimony score, underscoring that there are multiple optimal parsimony solutions. In contrast, the likelihood of a target block generally increases as the sum of the lengths of its parent blocks decreases, so the max likelihood DAG will not include edges that do not directly improve the score.

5 DISCUSSION

Our maximum parsimony and maximum likelihood reconstructions show some differences, both from each other and from the analysis of Jiang *et al.* (2007). In particular, we identify non-core duplicons and subsequences that are arguably as promiscuous within a clade as core duplicons.

There are several directions for future work. From a theoretical perspective, one can incorporate other types of operations into the probabilistic model, such as deletions and inversions which we have described in the parsimony setting (Kahn *et al.*, 2010), as well as single nucleotide mutations. Also, our method could be used to sample over the space of DAGs using a Markov Chain Monte Carlo strategy. From the perspective of applications, a more comprehensive analysis of genes or other elements in the newly identified core duplicons and core subsequences from our reconstruction is warranted, as is a further refinement of the clade annotation by analyzing the clade-induced subgraphs of the DAGs.

Funding: Career Award at the Scientific Interface from the Burroughs Wellcome Fund (to B.J.R.).

Conflict of Interest: none declared.

REFERENCES

Alkan,C. *et al.* (2009) Personalized copy number and segmental duplication maps using next-generation sequencing. *Nat. Genet.*, **41**, 1061–1067.

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

Blekhman,R. *et al.* (2009) Segmental duplications contribute to gene expression differences between humans and chimpanzees. *Genetics*, **182**, 627–630.

Chaudhuri,K. *et al.* (2006) On the tandem duplication-random loss model of genome rearrangement. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, ACM, New York, NY, USA, pp. 564–570.

El-Mabrouk,N. (2002) Reconstructing an ancestral genome using minimum segments duplications and reversals. *J. Comput. Syst. Sci.*, **65**, 442–464.

Ergun,F. *et al.* (2003) Comparing sequences with segment rearrangements. In *Proceedings FST TCS '03*, Vol. 2914, Springer, Berlin, pp. 222–234.

Jiang,Z. *et al.* (2007) Ancestral reconstruction of segmental duplications reveals punctuated cores of human genome evolution. *Nat. Genet.*, **39**, 1361–1368.

Kahn,C.L. and Raphael,B.J. (2008) Analysis of segmental duplications via duplication distance. *Bioinformatics*, **24**, i133–i138.

Kahn,C.L. and Raphael,B.J. (2009) A parsimony approach to analysis of human segmental duplications. *Pac. Symp. Biocomput.*, **14**, 126–137.

Kahn,C. *et al.* (2010) Efficient algorithms for analyzing segmental duplications with deletions and inversions in genomes. *Algorithms Mol. Biol.*, **5**, 11.

Lajoie,M. *et al.* (2007) Duplication and inversion history of a tandemly repeated genes family. *J. Comp. Bio.*, **14**, 462–478.

Marron,M. *et al.* (2004) Genomic distances under deletions and insertions. *Theor. Comput. Sci.*, **325**, 347–360.

McCaskill,J. (1990) The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, **29**, 1105–1119.

Price,A. *et al.* (2004) Whole-genome analysis of Alu repeat elements reveals complex evolutionary history. *Genome Res.*, **14**, 2245–2252.

Sankoff,D. (1999) Genome rearrangement with gene families. *Bioinformatics*, **15**, 909–917.