Pacific Symposium on Biocomputing 14:126-137 (2009)

A PARSIMONY APPROACH TO ANALYSIS OF HUMAN SEGMENTAL DUPLICATIONS

CRYSTAL L. KAHN* and BENJAMIN J. RAPHAEL†

Box 1910, Brown University
Department of Computer Science & Center for Computational Molecular Biology
Providence, RI 02912
U.S.A.
E-mail: {clkahn,braphael}@cs.brown.edu

Segmental duplications are abundant in the human genome, but their evolutionary history is not well-understood. The mystery surrounding them is due in part to their complex organization; many segmental duplications are mosaic patterns of smaller repeated segments, or duplicons. A two-step model of duplication has been proposed to explain these mosaic patterns. In this model, duplicons are copied and aggregated into primary duplication blocks that subsequently seed secondary duplications. Here, we formalize the problem of computing a duplication scenario that is consistent with the two-step model. We first describe a dynamic programming algorithm to compute the duplication distance between two strings. We then use this distance as the cost function in an integer linear program to obtain the most parsimonious duplication scenario. We apply our method to derive putative ancestral relationships between segmental duplications in the human genome.

1. Introduction

Mammalian genomes consist of many repetitive sequences. Many of these are insertions of (retro)transposons and are present in many copies, but longer segmental duplications, or low-copy repeats, are also common. Current estimates suggest that approximately 5% of the human genome consists of segmental duplications > 1 kb in length with $\geq 90\%$ sequence identity between copies[1]. Moreover, the pattern of segmental duplications in primate genomes is unusual: many of these duplications are interchromosomal[1] and are recent alterations in the human genome, occurring after the separation of the Old World monkey and great ape lineages. It is estimated that

*Work supported by a National Science Foundation Graduate Research Fellowship.
†Work supported by a Career Award at the Scientific Interface from the Burroughs Wellcome Fund.

Pacific Symposium on Biocomputing 14:126-137 (2009)

segmental duplications account for a significant fraction of the differences between humans and primate genomes, and it has been shown that some human segmental duplications contain novel gene families and genes under strong positive selection. In addition, segmental duplications are implicated in disease-causing rearrangements[2] and copy-number polymorphisms in human populations[1]. Thus, reconstructing the evolutionary history of segmental duplications is essential for understanding primate evolution and for ancestral genome reconstruction[3].

Despite the prominence of segmental duplications, both their evolutionary history and the mechanisms by which large segments of the genome are duplicated and transposed to other genomic loci are poorly understood. Studies of the human genome sequence revealed the surprising fact that many segmental duplications are complex mosaics of smaller duplicated pieces[4]. A **two-step model** of segmental duplication (reviewed in[1]) has been proposed to explain these mosaic patterns, but the convoluted nature of overlapping, interleaved duplicated material in the genome make segmental duplications refractory to traditional sequence analysis. Recently, Jiang El Ab.[5] examined the ancestral relationships between human segmental duplications, and identified "clades" of segmental duplications that share an abundance of repeated subsequences. However, their approach ignored the order and orientation of these repeated subsequences within the segmental duplications, and thus did not explicitly explain the mosaic organization of segmental duplications. While algorithms have been developed for various aspects of duplication analysis – such as the evolution of gene clusters[6,7,8], the analysis of whole-genome duplications[13,14], and the computation of reversal distance[9,10,11] and reconciliation of gene trees and species trees[12] in the presence of orthologous and paralogous genes – none of these methods are directly applicable to the analysis of the organization of segmental duplications. The problems of determining the evolutionary history of human segmental duplications and of validating the two-step model have not yet received a rigorous algorithmic treatment.

In this paper, we consider the problem of constructing a duplication scenario that is consistent with the two-step model of duplication, and that minimizes the total number of duplication operations. We formulate this problem as an integer linear program that is equivalent to the facility location problem, a classic problem in operations research. This formulation requires a measure of the minimum number of duplications to build a target string from a source string. We generalize the duplication distance algorithm introduced in [15] to compute the distance when both the source and

Pacific Symposium on Biocomputing 14:126-137 (2009)

target strings contain repeated characters, a requirement for applications to real genomic data. We apply our methods to the duplication blocks derived in [5] and discover a two-step duplication scenario in which 13 seed duplication blocks are first constructed and then duplicated to create 416 secondary duplication blocks.

2. Methods

Informally, given an ancestral genome that is devoid of duplications and a present-day genome rife with duplicated material, the problem is to compute a sequence of duplication events by which the ancestral genome is transformed into the present-day genome. In particular, we are interested in computing the simplest or *most-parsimonious* sequence of duplication events that accomplishes this transformation. In the next section, we describe a model of evolution that is consistent with the two-step model of duplication. Then in Section 2.2, we formulate the problem of computing the most-parsimonious duplication scenario as an integer linear program. Finally, in Section 2.3 we describe an algorithm to compute the duplication distance between a pair of signed strings, a distance used to define the cost function in the integer linear program. This algorithm generalizes the algorithm presented in[15], and an implementation is available upon request.

2.1. *Problem Formulation*

As described above, the segments of the present-day genome that contain duplicated material, hereafter *duplication blocks*, contain complex mosaic patterns of smaller segments, hereafter *duplicons*, that appear in multiplicity across the genome. We model both the ancestral and present-day genomes as signed strings of duplicons. We assume the present-day genome, which has incurred segmental duplications, is a superstring of the ancestral genome, and the duplication blocks are substrings of the present-day genome. See Figure 1.

In the two-step model of duplication, duplicons are copied from their ancestral loci and aggregated into larger, contiguous segments or *seed duplication blocks* during the first duplication phase. In the second phase, substrings of both the seed blocks and the ancestral genome are copied and then reinserted into the genome at disparate locations, creating *secondary duplication blocks*. We say that a seed duplication block *seeds* a secondary duplication block in the second phase if substrings of the seed block are used in the construction of the secondary block. See Figure 2a. Note that a seed block may seed multiple secondary duplication blocks but some seed

Pacific Symposium on Biocomputing 14:126-137 (2009)

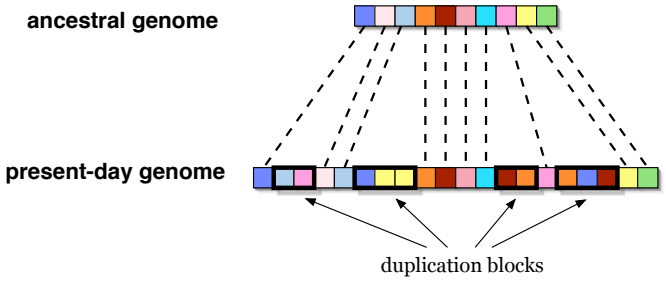

Figure 1. The present-day genome is a superstring of the ancestral genome. The duplicated material comprises *duplication blocks* which are maximal contiguous substrings of the present-day genome that were not part of the ancestral genome.

blocks may not seed any secondary blocks.

We make four simplifying assumptions about the two-step model of duplication:

(1) The ancestral genome contains exactly one copy of every duplicon.
(2) No other type of rearrangement operations – such as inversions of deletions – occur.
(3) The seed blocks are a subset of the duplication blocks observed in the present-day genome.
(4) Each secondary duplication block is seeded by exactly one seed duplication block.

Under these assumptions, we can summarize a two-step duplication scenario in a *two-step duplication tree*.

Definition 2.1. Given ancestral and present-day genomes, a *two-step duplication tree* is a tree of height three where the root is the ancestral genome and the descendants are the duplication blocks. Nodes at depth one (i.e. the children of the root) are the seed blocks created in the first phase of duplication, while nodes at depth two (i.e. children of seed blocks) are the secondary duplication blocks constructed from substrings of one seed block and of the ancestral genome. See Figure 2b.

For a given pair of ancestral and present-day genomes, a most-parsimonious two-step duplication tree is that which defines a partition of the duplication blocks into seed duplication blocks and secondary duplication blocks and defines the ancestral relationships between seed and secondary blocks such

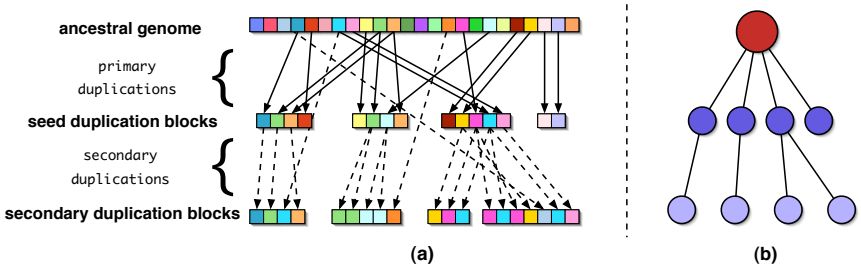

Figure 2. **(a)** The two-step model of duplication. Solid arrows indicate duplicons copied during the first phase of duplication. Dashed arrows indicate duplicons copied during the second phase of duplication. **(b)** The corresponding two-step duplication tree.

that the total number of duplication events needed to construct first the seed blocks and then the secondary blocks is minimum.

We define a duplication event as one *duplicate operation*:

Definition 2.2. A **duplicate operation**, $\delta_{s,t,p}(X)$, copies a substring $x_s \dots x_t$ of a source string X and pastes it into another string Z at position p. Specifically, if $X = x_1 \dots x_m$ and $Z = z_1 \dots z_n$, then $Z \circ \delta_{s,t,p}(X) = z_1 \dots z_{p-1} x_s \dots x_t z_p \dots z_n$.

In[15], we introduced a distance measure from a source string to a target string, called *duplication distance*.

Definition 2.3. The *duplication distance* from a source string X to a target string Y, denoted $D_X(Y)$, is the number of duplicate operations in the simplest (i.e. shortest) sequence of duplicate operations that transforms an initially empty string into the target Y by repeatedly inserting substrings of X.

The total duplication distance for a two-step duplication tree is the sum of the number of duplicate operations needed to build all the duplication blocks. We express the number of duplicate operations needed to build a seed block B_i from the ancestral genome $\mathcal{G}$ as $D_{\mathcal{G}}(B_i)$. Secondary duplication blocks are built from substrings of both its parent seed block and the ancestral genome. Thus, we express the number of duplicate operations needed to build a secondary block B_j from its parent seed block B_i and $\mathcal{G}$ as $D_{B_i \circ \mathcal{G}}(B_j)$, where $B_i \circ \mathcal{G}$ denotes the concatenation[a] of the strings B_i

[a]We insert a "dummy character" between B_i and $\mathcal{G}$ in the concatenate to avoid copying

Pacific Symposium on Biocomputing 14:126-137 (2009)

and $\mathcal{G}$.

We now state our problem.

`Problem:` Computing a Most-Parsimonious Two-Step Duplication Tree.
`Input:` The ancestral genome $\mathcal{G}$ and a set duplication blocks $B_1, \ldots, B_N$ from the present-day genome.
`Output:` The two-step duplication tree (Definition 2.1) with minimum total duplication distance.

2.2. *Computing the Two-Step Duplication Tree*

In this section, we show how to formulate the problem of constructing a most-parsimonious two-step duplication tree as an integer linear program (ILP).

A two-step duplication tree for a given ancestral genome and a set of duplication blocks is defined by a labeling of each of the N duplication blocks as either seed blocks or as secondary blocks. In addition to this labeling, we must also define for each secondary block which seed duplication block seeded it, i.e. which seed block is its parent in the tree.

A most-parsimonious two-step duplication tree is a solution of the following integer linear program.

$$\min_{U,V} \left[\sum_{i=1}^{N} (u_i \times D_{\mathcal{G}}(B_i)) + \sum_{i=1}^{N} \sum_{j=1}^{N} \left(v_{ij} \times D_{B_j \circ \mathcal{G}}(B_i) \right) \right] \tag{1}$$

such that

$$\sum_{j} v_{ij} = 1 \text{ for all } i \tag{2}$$

$$v_{ij} - u_j \geq 0 \text{ for all } i, j \tag{3}$$

$$u_i \in \{0, 1\} \text{ and } v_{ij} \in \{0, 1\} \tag{4}$$

The binary variables $U = [u_1, \ldots, u_N]$ and binary matrix $V = [v_{ij}]_{i,j=1}^{N}$ describe the topology of the duplication tree. The binary variable u_i indicates whether a duplication block B_i is labeled as a seed block and thus defines an edge in the tree from the root $\mathcal{G}$ to B_i. The binary variable v_{ij} indicates that secondary duplication block B_i is seeded by seed block B_j and thus block B_i is a child of B_j in the tree.

We note that this program is equivalent to a special case of the facility location problem, a classic NP-hard combinatorial optimization problem.

substrings across the boundary.

The input to the facility location problem is a set of customers and a set of potential facility sites. For each site, there is a cost associated with opening a facility, and for each site-customer pair, there is a cost associated with supplying that customer from a facility at that site. The objective is to minimize the total cost of opening facilities and supplying customers such that every customer is supplied by exactly one open facility. In the context of the two-step duplication tree, each duplication block is both a customer to be supplied and the site of a potential facility. Opening a facility at site B_i corresponds to classifying B_i as a seed duplication block. Supplying customer B_j from facility B_i corresponds to classifying B_j as a secondary block that is constructed from substrings of seed block B_i and the ancestral genome $\mathcal{G}$.

2.3. *Duplication Distance Algorithm*

In this section, we present a dynamic programming algorithm for computing duplication distance from signed source string X to signed target string Y. This algorithm generalizes the $O(|Y|^4)$ algorithm introduced in[15] for the special case where the source string X is *non-ambiguous* meaning that it contains at most one instance of each character. Here, we present an algorithm whose complexity is $O(|Y|^3|X|)$ for the general case where both X and Y are ambiguous. We assume that all the characters that appear in the target string also appear at least once in the source string.

Intuitively, if signed strings X and Y are "close" in duplication distance, they share common subsequences and thus there is duplication scenario that builds Y by repeated insertions of substrings of X (see Definitions 2.2 and 2.3).

We define some notation. Given a signed string $Y = y_1 y_2 \dots y_n$, let $Y_{s,t} = y_s y_{s+1} \dots y_t$ be the substring ranging between indices s and t, inclusive, for $1 \leq s \leq t \leq n$. Given strings X and Y, let $\mathcal{I}_{X,Y}(i) = \{j | x_j = y_i\}$. Finally, let $d(Y_{s,t}) = D_X(Y_{s,t})$ be the number of duplicate operations required to build the substring $Y_{s,t}$.

First, note that we can compute the duplication distance for disjoint substrings of the target Y independently, allowing us to recursively divide the target string into subproblems. Suppose we wish to compute the duplication distance for a substring $Y_{s,t}$ of Y. There are two possible cases to consider: (i) y_s was duplicated by itself as a substring of X of length one, or (ii) y_s was copied into Y as part of a larger substring of X. In the first case, we proceed recursively on $Y_{s+1,t}$. In the second case, we note that because X may be ambiguous, we must consider the possibility that the substring

Pacific Symposium on Biocomputing 14:126-137 (2009)

of X that was copied to produce the character at index s in Y might have started at any of the indices in $\mathcal{I}_{X,Y}(s)$. If a substring of X beginning at index i and having length greater than one is duplicated and inserted into Y, the character x_{i+1} must appear in Y at some index $j \in \mathcal{I}_{Y_{s+1,t},X}(i+1)$. Note, that although x_i was copied into Y at index s, it need not be the case that x_{i+1} appear at index $s+1$ in Y because subsequent duplicate operations may have inserted substrings into Y in between the characters x_i and x_{i+1}.

This observation leads to the following theorem.

Theorem 2.1. *Given a string $Y_{s,t}$ and indices $s \leq t$, $d(Y_{s,t})$ satisfies the following recurrence.*

$$d(Y_{s,t}) = \begin{cases} 1 & \text{if } s = t \text{ (base case),} \\ \min_{i \in \mathcal{I}_{X,Y}(s)} d_i(Y_{s,t}) & \text{otherwise,} \end{cases} \tag{5}$$

where

$$\begin{aligned} d_i(Y_{s,t}) &= \min\{a_{s,t}, b_{s,t}\}, \\ a_{s,t} &= 1 + d(Y_{s+1,t}), \\ b_{s,t} &= \min_{j \in \mathcal{I}_{Y_{s+1,t},X}(i+1)} \left(d(Y_{s+1,j-1}) + d_{i+1}(Y_{j,t})\right). \end{aligned}$$

Intuitively, $d_i(Y_{s,t})$ represents the number of duplicate operations needed to build $Y_{s,t}$ given that the character y_s resulted from the copying of a substring of X beginning at index i. The value of $d_i(Y_{s,t})$ can be expressed as the minimum of two values corresponding to the two possible cases described above. And the value $d(Y_{s,t})$ can then be computed by considering all possible indices i such that y_s is the same character as x_i.

The running time of the recurrence is bounded by the time to compute $d_i(Y_{s,t})$ for all possible values of i, s and t. For each substring $Y_{s,t}$, we compute $d_i(Y_{s,t})$ for each index $i \in \mathcal{I}_{X,Y}(s)$. This is bounded by $O(|X|)$ for all values of i, s and t. Because there are $O(|Y|^2)$ substrings $Y_{s,t}$, we compute $d_i(Y_{s,t})$ a total of $O(|Y|^2|X|)$ times. Finally, since the time to compute $d_i(Y_{s,t})$ for fixed i, s, and t is bounded by $|\mathcal{I}_{Y_{s+1,t},X}(s+1)| \in O(|Y|)$, the total running time is $O(|Y|^3|X|)$.

Note that it is straightforward to extend our algorithm to allow duplicate reversal operations on signed strings, i.e. duplicate operations in which the copied substring of X is inverted before being inserted into Y. We can accomplish by by defining $d(Y_{s,t})$ as the minimum of $d(Y_{s,t})$ and $d(-Y_{s,t})$ for all values of s and t, where $-Y_{s,t}$ denotes the reversal of $Y_{s,t}$.

Pacific Symposium on Biocomputing 14:126-137 (2009)

3. Results

We implemented our two-step duplication tree method to analyze the ancestry of segmental duplications in the human genome. We used data from[5] who identified 4,692 ancestral duplicons that appeared in complex mosaic arrangements within 437 duplication blocks in the human reference genome (hg17, May 2004). We enumerated the duplicons according to their ancestral location in the genome and aligned each duplication block to all duplicons using Nucmer[16] thereby representing each duplication block as a signed string in the alphabet of integers between -4692 and $+4692$. This process yielded a total of 429 duplication blocks, as 8 blocks could not be aligned to duplicons. We defined the ancestral genome $\mathcal{G}$ to be the string of duplicons in the order of their ancestral loci, i.e. $+1 + 2 \cdots + 4692$. We inserted dummy characters between successive duplicons whose ancestral loci were greater than 10kb apart.

For each ordered pair of duplication blocks B_i, B_j, we computed the distance $D_{B_i \circ \mathcal{G}}(B_j)$ using the algorithm presented in Section 2.3. We also computed, for every duplication block B_i, the distance $D_{\mathcal{G}}(B_i)$ from $\mathcal{G}$ to that block. Using these distances, we solved the ILP defined in (1) using CPLEX. Given the solution to the ILP, we labeled all blocks B_i such that the binary variables $u_i = 1$ as seed blocks. We labeled the remaining blocks as secondary blocks. For any pair of blocks B_i, B_j, if the binary variable $v_{ij} = 1$, we designated secondary block B_i to be a child of seed block B_j. The corresponding two-step duplication tree is shown in Figure 3.

This tree has 13 seed blocks, with one block (chr11:49122753-50286686) having 301 children. This duplication block is long, consisting of 2922 ancestral duplicons, but is not the longest in physical distance. Interestingly, this block is located near the centromere of chromosome 11, consistent with the hypothesis that duplicon seeding occurs in pericentromeric regions[1]. However, the large number of children of this block is likely an artifact that results from our restriction that secondary blocks are descended from exactly one seed block. A more parsimonious duplication scenario might have been that this block was itself formed by duplication from several other seed blocks.

4. Discussion and Future Directions

We formulated the problem of computing the most-parsimonious duplication scenario consistent with the two-step model, and showed how to solve this problem as an integer linear program that is equivalent to the facil-

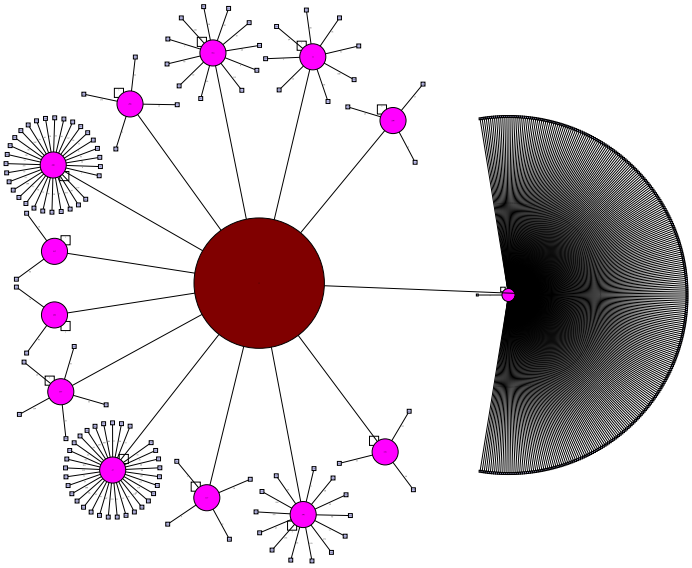

Figure 3. Most-parsimonious two-step duplication tree. The large node is the root and represents the ancestral genome. The thirteen children of the root are seed blocks that were constructed during the first step of duplication and are represented as medium-sized circles. Children of seed blocks were constructed during the second step and are represented as small squares. Starting with the seed block with the most children and proceeding clock-wise around the root, the seed blocks are: chr11:49122753-50286686, chr9:87954745-87985307, chr22:19790229-20121932, chr9:38462158-38620390, chr7:127698302-127893857, chr9:42280547-42512326, chr17:31799963-31889184, chr10:30678896-30730597, chr17:18219669-18692134, chr9:94148626-94280597, chr7:74410377-74802698, chr16:88650921-88822254, and chr17:34433993-34453309.

ity location problem. To our knowledge, this work is the first attempt to construct the ancestry of segmental duplications using parsimony methods. Our two-step duplication tree provides a set of putative seed blocks that are useful for further studies of the history of segmental duplications in the human genome.

Jiang et al.[5] used a different approach to analyze the ancestral relationships of segmental duplications. They represented each duplication block as a binary vector indicating the presence or absence of each duplicon and performed hierarchical clustering on these binary vectors. This analysis ignores the order and orientation of duplicons within a duplication block. We compared our 13 seed groups of duplication blocks (consisting of a

Pacific Symposium on Biocomputing 14:126-137 (2009)

seed block and its children) to the 24 "clades" of closely-related duplication blocks reported in Jiang et al. We found that some of our groups were enriched for blocks from a single clade. For example, two of the four members of seed group chr17:34433993-34453309 are shared with the 21 members of Clade M1 defined in[5], ($p = 0.017$ by hypergeometric test). However, in many instances the clusters produced by the two methods differed. To investigate further, we searched for instances where duplicon content and duplication distance diverged. For example, duplication block B_1 = chr7:74410377-74802698 is a seed block and is the parent of secondary block B_2 = chr16:5067675-5156097 in our analysis (Figure 3). In fact, the duplication distance $D_{B_1 \circ \mathcal{G}}(B_2) = 113$ is significantly lower than the average duplication distance to B_2 over all duplication blocks. However, the duplicon content of these two duplication blocks is sufficiently different so that the analysis in[5] placed them into different clades: chr7-2 and M1, respectively. This occurred because although these two duplication blocks did not share a large percentage of their duplicons, surprisingly they did share a few long, conserved subsequences of duplicons that appeared in the same order in both duplication blocks. The presence of conserved subsequences of duplicons suggests that deriving evolutionary relationships from duplicon content alone might be misleading in some cases. Conversely, we also found examples of duplication blocks that, while very similar in duplicon content, were "far" in terms of duplication distance indicating that the duplicons were not organized into the same order in both blocks. Duplication blocks B_3 = chr11:3365455-3580141 and B_4 = chr16:5229432-5285786 were both placed into clade M1 by[5], but were children of seed blocks chr11:49122753-50286686 and chr7:127698302-127893857, respectively, in our analysis.

Our method relied on several simplifying assumptions. These included: (1) duplicate operations were the only genome rearrangement events that occurred in constructing segmental duplications; (2) seed blocks that were constructed in the first phase of duplication still exist in an unaltered state in the present-day genome; (3) each secondary duplication block was descended from exactly one seed block. A more comprehensive analysis of segmental duplications will require the removal of some or all of these restrictions. In addition, we assumed that the work of identifying and delimiting the ancestral duplicons was already completed (e.g. by[5]). Incorporating duplication models into the annotation of duplication blocks themselves is an interesting avenue for future study. Finally, a similar two-step model involving extrachromosomal intermediates[17] has been proposed[18] to explain the extensive duplications of genomic segments in cancer genomes that

Pacific Symposium on Biocomputing 14:126-137 (2009)

also display a complex architecture[19]. It would be interesting to extend our methods to infer the sequence of duplications that occur in a cancer genome.

References

1. J. Bailey and E. Eichler, *Nat. Rev. Genet.* **7**, 552 (2006).
2. P. Stankiewicz and J. Lupski, *Trends Genet.* **18**, 74 (2002).
3. M. Blanchette, E. Green, W. Miller and D. Haussler, *Genome Res.* **14**, 2412 (2004).
4. J. Bailey, Z. Gu, R. Clark, K. Reinert, R. Samonte, S. Schwartz, M. Adams, E. Myers, P. Li and E. Eichler, *Science* **297**, 1003 (2002).
5. Z. Jiang, H. Tang, M. Ventura, M. F. Cardone, T. Marques-Bonet, X. She, P. A. Pevzner and E. E. Eichler, *Nature Genetics* **39**, 1361 (2007).
6. Y. Zhang, G. Song, T. Vinar, E. D. Green, A. C. Siepel and W. Miller, Reconstructing the evolutionary history of complex human gene clusters, in *RECOMB*, 2008.
7. O. Elemento, O. Gascuel and M.-P. Lefranc, *Mol Biol Evol* **19**, 278 (2002).
8. M. Lajoie, D. Bertrand, N. El-Mabrouk and O. Gascuel, *J. Comp. Bio.* **14**, 462 (2007).
9. N. El-Mabrouk, *J. Comput. Syst. Sci.* **65**, 442 (2002).
10. X. Chen, J. Zheng, Z. Fu, P. Nan, Y. Zhong, S. Lonardi and T. Jiang, *IEEE/ACM Trans. Comput. Biol. Bioinformatics* **2**, 302 (2005).
11. M. Marron, K. Swenson and B. Moret, Genomic distances under deletions and insertions (2003).
12. L. Arvestad, A.-C. Berglund, J. Lagergren and B. Sennblad, Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution, in *RECOMB '04: Proceedings of the eighth annual international conference on Research in computational molecular biology*, (ACM, New York, NY, USA, 2004).
13. N. El-Mabrouk and D. Sankoff, *SIAM J. Comput.* **32**, 754 (2003).
14. M. A. Alekseyev and P. A. Pevzner, *SIAM J. Comput.* **36**, 1748 (2007).
15. C. Kahn and B. Raphael, *Bioinformatics* **24**, i133 (2008).
16. S. Kurtz, A. Phillippy, A. Delcher, M. Smoot, M. Shumway, C. Antonescu and S. Salzberg, *Genome Biol.* **5** (2004).
17. B. E. Windle and G. M. Wahl, *Mutat Res* **276**, 199 (1992).
18. B. Raphael and P. Pevzner, *Bioinformatics* **20 Suppl 1**, i265 (2004).
19. B. Raphael, S. Volik, P. Yu, C. Wu, G. Huang, E. Linardopoulou, B. Trask, F. Waldman, J. Costello, K. Pienta, G. Mills, K. Bajsarowicz, Y. Kobayashi, S. Shivaranjani, P. Paris, Q. Tao, S. Aerni, R. Brown, A. Bashir, J. Gray, J. Cheng, P. de Jong, M. Nefedov, T. Ried, H. Padilla-Nash and C. Collins, *Genome Biol.* **9**, p. R59 (2008).