BMC Bioinformatics

PROCEEDINGS Open Access

Genomic distance under gene substitutions

Marília D V Braga[1*], Raphael Machado[1], Leonardo C Ribeiro[1], Jens Stoye[2]

From Ninth Annual Research in Computational Molecular Biology (RECOMB) Satellite Workshop on Comparative Genomics
Galway, Ireland. 8-10 October 2011

Abstract

Background: The distance between two genomes is often computed by comparing only the common markers between them. Some approaches are also able to deal with non-common markers, allowing the insertion or the deletion of such markers. In these models, a deletion and a subsequent insertion that occur at the same position of the genome count for two sorting steps.

Results: Here we propose a new model that sorts non-common markers with substitutions, which are more powerful operations that comprehend insertions and deletions. A deletion and an insertion that occur at the same position of the genome can be modeled as a substitution, counting for a single sorting step.

Conclusions: Comparing genomes with unequal content, but without duplicated markers, we give a linear time algorithm to compute the genomic distance considering substitutions and double-cut-and-join (DCJ) operations. This model provides a parsimonious genomic distance to handle genomes free of duplicated markers, that is in practice a lower bound to the real genomic distances. The method could also be used to refine orthology assignments, since in some cases a substitution could actually correspond to an unannotated orthology.

Background

The genomic distance is often computed taking into consideration only the common markers, that occur in both genomes [1-3]. Approaches to deal with unique markers (that occur in only one genome) also exist, but usually allowing only insertions or deletions of these markers. Insertions and deletions can be shortly called *indels*. In [4], the operations allowed are inversions and indels, while the models given in [5] and [6] consider indels and the double cut and join (DCJ) operation [7], that is able to represent most large scale mutation events in genomes, such as inversions, translocations, fusions and fissions. The mentioned approaches assign the same weight to all rearrangement operations, including indels, regardless of the size of the affected regions and the particular types of the operations. A drawback in these models is that, if a deletion and a subsequent insertion occur at the same position of the genome, the cost is the same as a deletion and an insertion in different positions.

In the present work we propose a more parsimonious model in which, instead of deleting or inserting, we allow the substitution of unique markers between two genomes, as illustrated in Figure 1. We do not suggest that a substitution occurs in a precise moment in evolution, but instead it represents a region that underwent continuous mutations (duplications, losses and gene mutations), so that a group of genes is transformed into a different group of genes (either of which may also be empty, allowing a substitution to represent an insertion or a deletion). Other studies also represent continuous mutations as a rearrangement event [8,9]. By minimizing substitutions we are able to establish a relation between indels that could have occurred in the same position of the compared genomes, identifying genomic regions that could be subject to these continuous mutations. Observe that we suggest that such regions have a common evolutionary origin. We develop a method to count the minimum number of substitutions that could have occurred, by assigning the same weight to substitutions and to the

* Correspondence: mdbraga@inmetro.gov.br
[1]Instituto Nacional de Metrologia, Qualidade e Tecnologia, Duque de Caxias, 25250-020, Brazil
Full list of author information is available at the end of the article

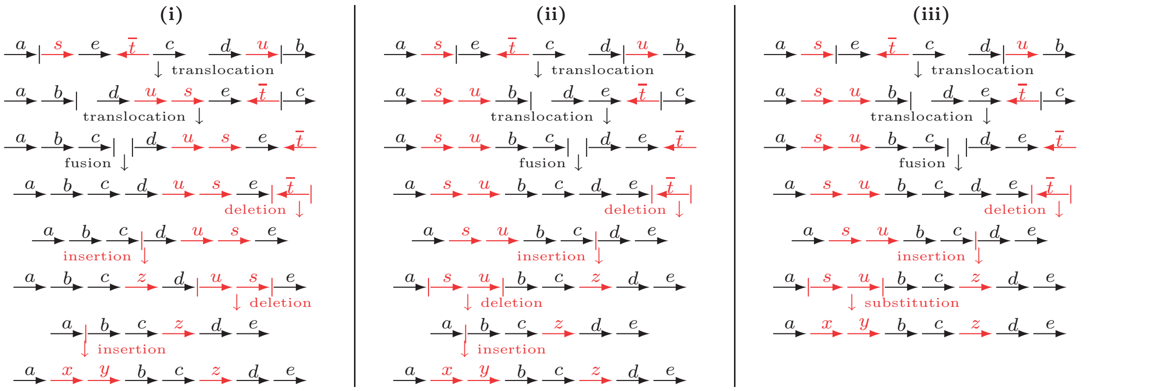

Figure 1 (i) An optimal sorting scenario with DCJ operations and indels. (ii) An optimal sorting scenario with DCJ operations and indels in which the last two operations occur in the same position of the genome, between markers *a* and *b*. (iii) A more parsimonious alternative to the deletion of consecutive markers *s* and *u* and the insertion of consecutive markers *x* and *y* would be the substitution of *s* and *u* by *x* and *y*.

other operations, similarly to the approaches that handle indels.

We analyze genomes with unequal content, but without duplicated markers and extend the results given in [6] to develop a linear time algorithm that exactly computes the genomic distance with substitutions and DCJ operations. The objective of this model is to provide a parsimonious genomic distance to handle genomes free of duplicated markers, that in practice is a lower bound to the real genomic distances. In the present work, we do not study algorithms to generate parsimonious sorting scenarios. Nevertheless, in the analysis of the evolution of human chromosomes X and Y, we manually obtain a parsimonious evolutionary scenario under our model, that is coherent with the results given in [10].

In the remainder of this section we introduce some concepts given in [1] and [6] and define the operation that substitutes markers in a genome - these are the basis of the method that we will present here.

Preliminaries

In the present study duplicated markers are not allowed. Given two genomes A and B, possibly with unequal content, we denote by $\mathcal{G}$ the "reduced" genome [4], that is the set of markers that occur once in A and once in B. Moreover, the set $\mathcal{A}$ contains the markers that occur only in A and the set $\mathcal{B}$ contains the markers that occur only in B. The markers in sets $\mathcal{A}$ and $\mathcal{B}$ are also called *unique markers*. Observe that the sets $\mathcal{G}$, $\mathcal{A}$ and $\mathcal{B}$ are disjoint.

A genome is possibly composed of linear and circular chromosomes. Each marker g in a genome is a DNA fragment and is represented by the symbol g, if it is read in direct orientation, or by the symbol $\overline{g}$, if it is read in reverse orientation. An example of a pair of genomes is given in Figure 2.

In the following we adopt definitions which we have given in [6] (some of them are generalizations of concepts introduced by Bergeron *et al.* [1]).

$\mathcal{G}$ -adjacencies

Each one of the two ends of a linear chromosome is called a *telomere* and is represented by the symbol ∘. For each marker $g \in \mathcal{G}$, denote its two extremities by g^t (tail) and g^h (head). A $\mathcal{G}$ *-adjacency* in genome A (respectively in genome B) is in general a linear string $v = \gamma_1 \ell \gamma_2$, such that γ_1 and γ_2 are telomeres or extremities of markers of $\mathcal{G}$ and ℓ, the string composed of the markers that are between γ_1 and γ_2 in A (respectively in B), contains no marker that also belongs to $\mathcal{G}$. The string ℓ is said to be the *label* of v, and the extremities γ_1 and γ_2 are said to be $\mathcal{G}$ *-adjacent*. If ℓ is a non-empty string, v is said to be *labeled*, otherwise v is said to be *clean*.

A $\mathcal{G}$ -adjacency $\gamma_1 \ell \gamma_2$ can also be represented by $\mathcal{G}$. Furthermore, $\circ\ell\circ$ represents a linear chromosome

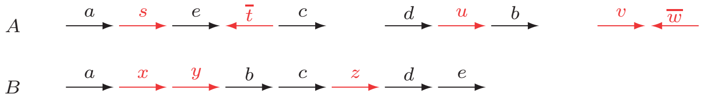

Figure 2 For genomes A, composed of three linear chromosomes, and *B*, composed of one single chromosome, we have $\mathcal{G} = \{a, b, c, d, e\}$, $\mathcal{A} = \{s, t, u, v, w\}$ and $\mathcal{B} = \{x, y, z\}$.

composed only of markers that are not in $\mathcal{G}$. In the same way, a $\mathcal{G}$ -adjacency given by a label ℓ corresponds to a whole circular chromosome composed only of markers that are not in $\mathcal{G}$. This is the only case of a $\mathcal{G}$ -adjacency in which we have a circular instead of a linear string.

Two genomes A and B can then be represented by the sets $V_{\mathcal{G}}(A)$ and $V_{\mathcal{G}}(B)$, containing their $\mathcal{G}$ -adjacencies. For the two genomes in Figure 2, we have $V_{\mathcal{G}}(B) = \{\circ a^t, a^h xyb^t, b^h c^t, c^h zd^t, d^h e^t, e^h \circ\}$, $V_{\mathcal{G}}(B) = \{\circ a^t, a^h xyb^t, b^h c^t, c^h zd^t, d^h e^t, e^h \circ\}$ and $V_{\mathcal{G}}(B) = \{\circ a^t, a^h xyb^t, b^h c^t, c^h zd^t, d^h e^t, e^h \circ\}$.

The DCJ operation

A *cut* performed on a genome A separates two adjacent markers of A. A cut affects a $\mathcal{G}$ -adjacency v of $V_{\mathcal{G}}(A)$ as follows: if v is linear, the cut is done between two symbols of v, creating two open ends in two separate linear strings; if v is circular, the cut creates two open ends in one linear string. A *double-cut and join* or DCJ applied on a genome A is the operation that generally performs two cuts in $V_{\mathcal{G}}(A)$, creating four open ends, and joins these open ends in a different way. A DCJ operation can correspond to several rearrangement events, such as an inversion, a translocation, a fusion, or a fission [7].

We represent by $(\{\gamma_1\ell_1|\ell_4\gamma_4 , \gamma_3\,\ell_3 \mid \ell_2\gamma_2\} \rightarrow \{\gamma_1\ell_1|\,\ell_2\,\gamma_2, \gamma_3\,\ell_3|\ell_4\,\gamma_4\})$ a DCJ applied on $\gamma_1\ell_1\ell_4\gamma_4$ and $\gamma_3\ell_3\ell_2\gamma_2$, that creates $\gamma_1\ell_1\ell_2\gamma_2$ and $\gamma_3\ell_3\ell_4\gamma_4$. Observe that one or more extremities among γ_1, γ_2, γ_3 and γ_4 can be equal to $\circ$ (a telomere), as well as one or more labels among ℓ_1, ℓ_2, ℓ_3 and ℓ_4 can be equal to ε (the empty string). Particular cases include circular adjacencies and are described in [6].

Adjacency graph and the DCJ distance

The *adjacency graph* $AG(A, B)$ [*1*] is the bipartite graph that has a vertex for each $\mathcal{G}$ -adjacency in $V_{\mathcal{G}}(A)$ and a vertex for each $\mathcal{G}$ -adjacency in $V_{\mathcal{G}}(B)$. Then, for each $g \in \mathcal{G}$, we have one edge connecting the vertex in $V_{\mathcal{G}}(A)$ and the vertex in $V_{\mathcal{G}}(B)$ that contain g^h and one edge connecting the vertex in $V_{\mathcal{G}}(A)$ and the vertex in $V_{\mathcal{G}}(B)$ that contain g^t.

The connected components of the graph $AG(A, B)$ are cycles and paths that alternate vertices in $V_{\mathcal{G}}(A)$ and $V_{\mathcal{G}}(B)$. A path that has one endpoint in $V_{\mathcal{G}}(A)$ and the other in $V_{\mathcal{G}}(B)$ is called an *AB-path*. In the same way, both endpoints of an *AA-path* are in $V_{\mathcal{G}}(A)$, as well as both endpoints of a *BB-path* are in $V_{\mathcal{G}}(B)$. Furthermore, $AG(A, B)$ can have two extra types of components: each $\mathcal{G}$ -adjacency that corresponds to a linear (respect. circular) chromosome is a *linear* (respect. *circular*) *singleton*. Linear singletons are particular cases of AA-paths and BB-paths. An example of an adjacency graph is given in Figure 3.

The number of AB-paths in $AG(A, B)$ is always even and a DCJ operation can be of three types [1,6]: *optimal* when it either increases the number of cycles by one, or the number of AB-paths by two; *neutral* when it does not affect the number of cycles and AB-paths; or *counter-optimal* when it either decreases the number of cycles by one, or the number of AB-paths by two.

Singletons, AB-paths composed of one single edge, and cycles composed of two edges are said to be *DCJ-sorted*. Longer paths and cycles are said to be *DCJ-unsorted*. The procedure of using DCJ operations to turn $AG(A, B)$ into DCJ-sorted components is called *DCJ-sorting* of A into B. The *DCJ distance* of A and B, denoted by $d_{DCJ}(A, B)$, corresponds to the minimum number of steps required to do a DCJ-sorting of A into B and can be easily obtained:

Theorem 1 ([1])*Given two genomes A and B without duplicated markers, we have* $d_{DCJ}(A,B) = |\mathcal{G}| - c - \frac{b}{2}$, *where* $\mathcal{G}$ *is the set of common markers between A and B, and c and b are the number of cycles and of AB-paths in AG(A, B).*

Runs of unique markers

Given a component C of AG (A, B), we can obtain a string $\ell(C)$ by the concatenation of the labels of the $\mathcal{G}$ -adjacencies of C in the order in which they appear. Cycles, AA-paths and BB-paths can be read in any direction, but AB-paths should always be read from A to B. If C is a cycle and has labels in both genomes A and B, we should start to read in a labeled $\mathcal{G}$ -adjacency v of A, such that the first labeled vertex before v is a

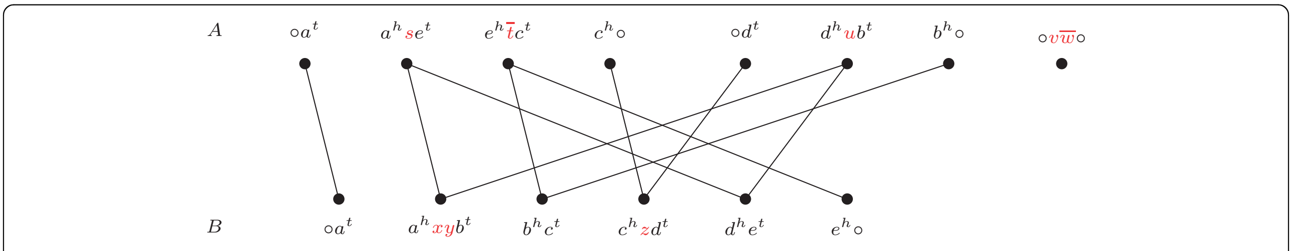

Figure 3 For genomes *A* and *B*, the adjacency graph contains one cycle, two *AA*-paths (one is a linear singleton) and two *AB*-paths.

$\mathcal{G}$ -adjacency in *B*; otherwise *C* has labels in at most one genome and we can start anywhere. Each maximal substring of $\ell(C)$ composed only of markers in $\mathcal{A}$ (respectively in $\mathcal{B}$ is called an $\mathcal{A}$ *-run* (respectively a $\mathcal{B}$ *-run*). Each $\mathcal{A}$ -run or $\mathcal{B}$ -run can be simply called *run*[6]. A component composed only of clean $\mathcal{G}$ -adjacencies has no run and is said to be *clean*, otherwise the component is *labeled*. We denote by $\Lambda(C)$ the number of runs in a component *C*. A path can have any number of runs, while a cycle has zero, one, or an even number of runs. Figure 4 shows a *BB*-path with 4 runs.

Substitutions

The unique markers in $\mathcal{A}$ and $\mathcal{B}$ are represented in *AG* (*A*, *B*) as labels and singletons and, in order to sort *A* into *B*, they also have to be considered. Here we propose a model in which only the following operation can be applied to unique markers. A *substitution* is an operation that affects the label of one single $\mathcal{G}$ -adjacency, by substituting contiguous markers in this label.

Consider the labels ℓ_1 and ℓ_2, where $|\ell_1| = m$ and $|\ell_2| = n$. The substitution of ℓ_1 by ℓ_2 in a $\mathcal{G}$ -adjacency is represented by $(\gamma_1\ell_3|\ell_1|\ell_4\gamma_2 \rightarrow \gamma_1\ell_3|\ell_2|\ell_4\gamma_2)$ (for better reading in our notation we omit the curly set brackets for singleton sets). One or both extremities among γ_1 and γ_2 can be equal to $\circ$ (a telomere), as well as one or both labels among ℓ_3 and ℓ_4 can be equal to ε (the empty string). The substitution of ℓ_1 by ℓ_2 in a circular singleton is represented by $(|\ell_1|\ell_3| \rightarrow |\ell_2|\ell_3|)$. Observe that at most one chromosome can be entirely substituted at once (but we do not allow the substitution of a linear by a circular chromosome and *vice-versa*). Moreover, if $m = 0$, we have an *insertion* of *n* contiguous markers. On the other hand, if $n = 0$, we have a *deletion* of *m* contiguous markers. Thus, insertions and deletions, also called *indels*, are special cases of substitutions.

The *DCJ-substitution distance* of *A* and *B*, denoted by $d_{DCJ}^{sb}(A,B)$, is the minimum number of DCJs and substitutions required to transform *A* into *B*. Since substitutions include indels, $d_{DCJ}^{sb}(A,B)$ is upper bounded by the *DCJ-indel distance*, the minimum number of DCJ and indel operations required to transform *A* into *B*, that can be computed in linear time [6]. In the present work we give an approach to exactly compute $d_{DCJ}^{sb}(A,B)$ also in linear time.

Results and discussion

The main result of the present study is an exact formula to compute the DCJ-substitution distance in linear time. We achieve this formula by developing the substitution-potential of two genomes, a property that allows us to obtain a good upper bound to the genomic distance with DCJ operations and substitutions. Then we show how some special DCJ operations reduce the overall number of substitutions and obtain the exact formula. Although the objective of this model is to provide a parsimonious genomic distance, that in practice is a lower bound to real distances, we run some experiments on data from human X and Y chromosomes and obtained a parsimonious sorting scenario that is coherent with the results available in the literature. We also observe that the DCJ-substitution method could be used to refine orthology assignments.

The substitution-potential

Observe that a $\mathcal{G}$ -adjacency with a non-empty label ℓ can be cut in at least two different positions, either before or after ℓ. Since the position of the cut does not change the effect of the DCJ on $d_{\mathrm{DCJ}}(A, B)$, we can choose to cut at positions that allow the concatenation of the labels of the original $\mathcal{G}$ -adjacencies. As a consequence, a set of labels of one genome can be *accumulated* with DCJ operations. In particular, when we apply optimal DCJs on only one component of the adjacency graph, we can accumulate an entire run in a single $\mathcal{G}$ -adjacency:

Proposition 1 ([6])*A run can be entirely accumulated in the label of one single* $\mathcal{G}$ *-adjacency with optimal DCJ operations.*

Given a DCJ operation ρ, let Λ_0 and Λ_1 be, respectively, the number of runs in *AG* (*A*, *B*) before and after ρ. We define $\Delta\Lambda(\rho) = \Lambda_{\ 1} - \Lambda_{\ 0}$.

Proposition 2 ([6])*Given any DCJ operation* ρ, *we have* $\Delta\Lambda(\rho) \geq -2$.

In order to obtain the exact formula for the DCJ-substitution distance, we will first analyze the components of the adjacency graph separately. Given two genomes *A* and *B* and a component $C \in AG$ (*A*, *B*), we denote by $d_{DCJ}(C)$ the minimum number of DCJ operations required to do a separate DCJ-sorting in *C*, applying

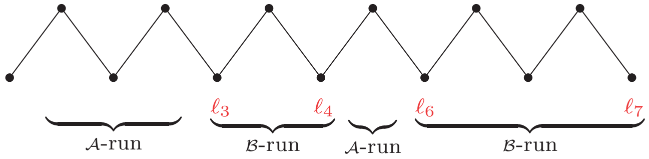

Figure 4 A *BB*-path with 4 runs. Only the labels of the $\mathcal{G}$ -adjacencies are represented.

DCJs on vertices of C (or vertices that result from DCJs applied on vertices that were in C). It is possible to do a separate DCJ-sorting using only optimal DCJs in any component of AG (A, B), thus, in other words, $d_{DCJ}(A, B) = \Sigma_{C\in AG(A,B)} d_{DCJ}(C)$ [2]. In [6] we have already defined the *indel-potential* of a component, denoted by $\lambda(C)$, that is the minimum number of runs that we can obtain by DCJ-sorting C with optimal DCJ operations only, and can be computed with the formula given in the next proposition.

Proposition 3 ([6])*Given a component C in AG*(A, B), *we have* $\lambda(C) = \left\lceil \frac{\Lambda(C)+1}{2} \right\rceil$, *if* $\Lambda(C) \geq 1$. *Otherwise* $\lambda(C) = 0$.

Similarly, here we denote by $\sigma(C)$ the *substitution-potential* of a component C, that is the minimum number of substitutions that we can obtain by DCJ-sorting C with optimal DCJ operations only. In order to find a formula to compute $\sigma(C)$, we first obtain a stronger version of Proposition 1 where not only the labels of a run are accumulated into a single $\mathcal{G}$ -adjacency, but pairs of consecutive runs are accumulated into adjacent $\mathcal{G}$ -adjacencies (that are $\mathcal{G}$ -adjacencies connected by a single edge in the adjacency graph).

Proposition 4 ([6])*If* $\gamma_1\gamma_2$ *is a clean* $\mathcal{G}$ *-adjacency in a DCJ-unsorted component C of AG*(A, B), *such that neither* γ_1 *nor* γ_2 *are telomeres, then it is always possible to extract a clean cycle from C with an optimal DCJ operation.*

Proposition 5 *Two consecutive runs in a component C can be entirely accumulated into the labels of two adjacent* $\mathcal{G}$ *-adjacencies of C with optimal DCJs.*

Proof: By Proposition 1 we assume that two consecutive runs of C are accumulated into $\mathcal{G}$ -adjacencies v_A and v_B. If v_A and v_B are not adjacent, there are only clean $\mathcal{G}$ -adjacencies between v_A and v_B in C. By Proposition 4, we can apply optimal DCJs to extract clean cycles until v_A and v_B are adjacent.

Pairs of consecutive runs that are accumulated into adjacent $\mathcal{G}$ -adjacencies can be extracted into a labeled DCJ-sorted component, that can be sorted with one substitution. Observe that minimizing the number of pairs of consecutive runs is equivalent to minimizing the total number of runs. Hence, we can determine the substitution-potential from the indel-potential.

Proposition 6 *Given a component C in AG* (A, B), *we have* $\sigma(C) = \left\lceil \frac{\Lambda(C)+1}{4} \right\rceil$, *if* $\Lambda(C) \geq 1$. *Otherwise* $\sigma(C) = 0$.

Proof: By Proposition 5 we can assume that the runs of C are accumulated into pairs of adjacent $\mathcal{G}$ -adjacencies. By Proposition 3, we can obtain $\left\lceil \frac{\Lambda(C)+1}{2} \right\rceil$ runs doing a separate DCJ-sorting in C with optimal DCJs. Moreover, these optimal DCJs can be done in such a way that pairs of runs that were accumulated into adjacent $\mathcal{G}$ -adjacencies remain in these adjacent $\mathcal{G}$ -adjacencies. Since each one of these pairs can be sorted with one substitution, the substitution-potential of C is equal to the number of pairs of labeled adjacent $\mathcal{G}$ -adjacencies, which is:

$$\sigma(C) = \left\lceil \frac{\frac{\Lambda(C)+1}{2}}{2} \right\rceil = \left\lceil \frac{\Lambda(C)+1}{4} \right\rceil.$$

The formulas to compute $\lambda(C)$ and $\sigma(C)$, given in Propositions 3 and 6 above, are indeed very similar. Consequently, many of the results obtained in [6] can be adapted to the new substitution-potential. Let σ_0 and σ_1 be, respectively, the sums of the number σ for the components of the adjacency graph before and after a DCJ operation ρ. We then define $\Delta\sigma(\rho) = \sigma_1 - \sigma_0$. Furthermore, let $\Delta_{dcj}(\rho)$ be respectively 0, +1 and +2 depending whether ρ is optimal, neutral or counter-optimal. We also define $\Delta d(\rho) = \Delta_{dcj}(\rho) + \Delta\sigma(\rho)$.

Proposition 7 *Given a DCJ operation* ρ *acting on a single component, we have* $\Delta d(\rho) \geq +2$ *if* ρ *is counter-optimal, or* $\Delta d(\rho) \geq 0$ *if* ρ *is neutral.*

We denote by $d_{DCJ}^{sb}(C)$ the minimum number of DCJs and substitutions required to sort separately a component C of AG (A, B). The definition of σ and Proposition 7 guarantee that $d_{DCJ}^{sb}(C) = d_{DCJ}(C) + \sigma(C)$.

Observe that, if C is a singleton in the adjacency graph, $d_{DCJ}^{sb}(C) = 1$, corresponding to the insertion or the deletion of the whole chromosome. We do not allow the substitution of a linear by a circular singleton and *vice-versa*. However, each pair composed by a singleton in genome A and a singleton in genome B (such that both are linear or both are circular) can be sorted with one single substitution, which saves one sorting step per pair. Let P_L and P_C be, respectively, the maximum number of disjoint pairs of linear and circular singletons in the adjacency graph. Together with the DCJ-substitution distance per component, these numbers give a good upper bound for $d_{DCJ}^{sb}(A, B)$:

Lemma 1 *Given two genomes A and B without duplicated markers, we have:*

$$d_{DCJ}^{sb}(A, B) \leq d_{DCJ}(A, B) + \sum_{C\in AG(A,B)} \sigma(C) - P_L - P_C.$$

The formula given by Lemma 1 above corresponds to the exact distance for a particular set of genomes. Given a $\mathcal{G}$ -adjacency $\gamma\ell\circ$ of a genome A such that $\gamma\neq\circ$, then γ is said to be a *tail* of a linear chromosome in A. Two genomes are *co-tailed* if their sets of tails are equal (this includes two genomes composed only of circular chromosomes).

Theorem 2 *Given two co-tailed genomes A and B without duplicated markers, we have:*

$$d_{DCJ}^{sb}(A, B) = d_{DCJ}(A, B) + \sum\nolimits_{C\in AG(A,B)} \sigma(C) - P_L - P_C.$$

However, for non co-tailed genomes the use of DCJs applied to two components of the adjacency graph can lead to a shorter sequence of operations sorting one genome into another, as we will see in the next section.

The DCJ-substitution distance

Recall that $\Delta\sigma(\rho) = \sigma_1 - \sigma_0$, where σ_0 and σ_1 are the sums of the number σ for the components of the adjacency graph before and after ρ. A DCJ operation ρ that acts on two components of the adjacency graph is called *recombination.*

Proposition 8 *Given any recombination* ρ, *we have* $\Delta\sigma(\rho) \geq -2$.

Proof: Only the recombinations that decrease or do not change the number of runs ($\Delta\Lambda \leq 0$) have to be analyzed (we can not have $\Delta\sigma \leq -1$ if the number of runs increases). Consider the recombination of two paths with i and j runs, that result in two new paths with i' and j' runs. The best we can have is when i and j are multiples of 4, i' and j' are multiples of 4 minus 1 and $\Delta\Lambda = -2$, that gives: $\sigma_1 = \left\lceil \frac{i'+1}{4} \right\rceil + \left\lceil \frac{j'+1}{4} \right\rceil = \frac{i'+j'+2}{4} = \frac{i+j}{4} = \frac{i}{4} = \frac{j}{4} = \left\lceil \frac{i+1}{4} \right\rceil - 1 + \left\lceil \frac{j+1}{4} \right\rceil - 1 = \sigma_0 - 2$. The analysis of recombinations involving cycles is analogous.

All recombinations involving at least one cycle are counter-optimal and any counter-optimal recombination has $\Delta d \geq 0$, thus only path recombinations can have $\Delta d \leq -1$. The following definitions are similar to those given in [6], except that here we have a larger number of labeled path types.

Consider an integer $i \geq 0$. For a second integer $k \in \{1, 3\}$, let $\mathcal{A} + k$ (respectively $\mathcal{B} + k$) be a sequence with odd $4i + k$ runs, starting and ending with an $\mathcal{A}$ -run (respectively $\mathcal{B}$ -run). Similarly for $k \in \{2, 4\}$, let $\mathcal{AB} + k$ (respectively $\mathcal{BA} + k$), be a sequence with even $4i + k$ runs, starting with an $\mathcal{A}$ -run (respectively $\mathcal{B}$ -run) and ending with a $\mathcal{B}$ -run (respectively $\mathcal{A}$ -run). An empty sequence (with no run) is represented by ε. Then each one of the notations AA_ε, $AA_{\mathcal{B}+1}$, $AA_{\mathcal{B}+1}$, $AA_{\mathcal{AB}+2}$, $AA_{\mathcal{A}+3}$, $AA_{\mathcal{B}+3}$, $AA_{\mathcal{AB}+4}$, BB_ε, $BB_{\mathcal{A}+1}$, $BB_{\mathcal{B}+1}$, $BB_{\mathcal{AB}+2}$, $BB_{\mathcal{A}+3}$, $BB_{\mathcal{B}+3}$, $BB_{\mathcal{AB}+4}$, AB_ε, $AB_{\mathcal{A}+1}$, $AB_{\mathcal{B}+1}$, $AB_{\mathcal{AB}+2}$, $AB_{\mathcal{BA}+2}$, $AB_{\mathcal{B}+3}$, $AB_{\mathcal{B}+3}$, $AB_{\mathcal{AB}+4}$ and $AB_{\mathcal{BA}+4}$ represents a particular type of path (AA, BB or AB) with a particular structure of runs (ε, $\mathcal{A}+1$, $\mathcal{B}+1$, $\mathcal{AB}+2$, $\mathcal{BA}+2$, $\mathcal{B}+3$, $\mathcal{B}+3$, $\mathcal{AB}+4$, or $\mathcal{BA}+4$).

The components on which the cuts are applied are called *sources* and the components obtained after the joinings are called *resultants* of the recombination. The complete set of recombinations with $\Delta d \leq -1$ is given in Table 1. In Table 2 we also list recombinations with $\Delta d = 0$ that create at least one source of recombinations of Table 1. We denote by $\bullet$ an AB-path that can not be a source in Tables 1 and 2, such as AB_ε, $AB_{\mathcal{A}+1}$, $AB_{\mathcal{B}+1}$, $AB_{\mathcal{BA}+2}$, $AB_{\mathcal{BA}+2}$, $AB_{\mathcal{A}+3}$ and $AB_{\mathcal{B}+3}$.

Table 1 Path recombinations that have Δ*d* ≤ –1 and allow the best reuse of the resultants.

sources	resultants	Δσ	Δ_{dcj}	Δd
$AA_{\mathcal{AB}+4} + BB_{\mathcal{AB}+4}$	$\cdot + \cdot$	–2	0	–2
$AA_{\mathcal{AB}+4} + AA_{\mathcal{AB}+4}$	$AA_{\mathcal{A}+3} + AA_{\mathcal{B}+3}$	–2	+1	–1
$BB_{\mathcal{AB}+4} + BB_{\mathcal{AB}+4}$	$BB_{\mathcal{A}+3} + BB_{\mathcal{B}+3}$	–2	+1	–1
$AA_{\mathcal{AB}+4} + AB_{\mathcal{AB}+4}$	$\bullet + AA_{\mathcal{A}+3}$	–2	+1	–1
$AA_{\mathcal{AB}+4} + AB_{\mathcal{BA}+4}$	$\bullet + AA_{\mathcal{B}+3}$	–2	+1	–1
$BB_{\mathcal{AB}+4} + AB_{\mathcal{AB}+4}$	$\bullet + BB_{\mathcal{B}+3}$	–2	+1	–1
$BB_{\mathcal{AB}+4} + AB_{\mathcal{BA}+4}$	$\bullet + BB_{\mathcal{A}+3}$	–2	+1	–1
$AA_{\mathcal{A}+1} + BB_{\mathcal{AB}+4}$	$\bullet + AB_{\mathcal{AB}+4}$	–1	0	–1
$AA_{\mathcal{B}+1} + BB_{\mathcal{AB}+4}$	$\bullet + AB_{\mathcal{BA}+4}$	–1	0	–1
$AA_{\mathcal{AB}+4} + BB_{\mathcal{A}+1}$	$\bullet + AB_{\mathcal{BA}+4}$	–1	0	–1
$AA_{\mathcal{AB}+4} + BB_{\mathcal{B}+1}$	$\bullet + AB_{\mathcal{AB}+4}$	–1	0	–1
$AA_{\mathcal{AB}+2} + BB_{\mathcal{AB}+4}$	$\cdot + \cdot$	–1	0	–1
$AA_{\mathcal{AB}+4} + BB_{\mathcal{AB}+2}$	$\cdot + \cdot$	–1	0	–1
$AA_{\mathcal{AB}+2} + BB_{\mathcal{AB}+2}$	$\cdot + \cdot$	–1	0	–1
$AA_{\mathcal{A}+3} + BB_{\mathcal{AB}+4}$	$\cdot + \cdot$	–1	0	–1
$AA_{\mathcal{B}+3} + BB_{\mathcal{AB}+4}$	$\cdot + \cdot$	–1	0	–1
$AA_{\mathcal{AB}+4} + BB_{\mathcal{A}+3}$	$\cdot + \cdot$	–1	0	–1
$AA_{\mathcal{AB}+4} + BB_{\mathcal{B}+3}$	$\cdot + \cdot$	–1	0	–1
$AA_{\mathcal{A}+1} + BB_{\mathcal{A}+1}$	$\cdot + \cdot$	–1	0	–1
$AA_{\mathcal{B}+1} + BB_{\mathcal{B}+1}$	$\cdot + \cdot$	–1	0	–1
$AA_{\mathcal{A}+1} + BB_{\mathcal{AB}+2}$	$\cdot + \cdot$	–1	0	–1
$AA_{\mathcal{B}+1} + BB_{\mathcal{AB}+2}$	$\cdot + \cdot$	–1	0	–1
$AA_{\mathcal{AB}+2} + BB_{\mathcal{A}+1}$	$\cdot + \cdot$	–1	0	–1
$AA_{\mathcal{AB}+2} + BB_{\mathcal{B}+1}$	$\cdot + \cdot$	–1	0	–1
$AA_{\mathcal{A}+1} + BB_{\mathcal{A}+3}$	$\cdot + \cdot$	–1	0	–1
$AA_{\mathcal{B}+1} + BB_{\mathcal{B}+3}$	$\cdot + \cdot$	–1	0	–1
$AA_{\mathcal{A}+3} + BB_{\mathcal{A}+1}$	$\cdot + \cdot$	–1	0	–1
$AA_{\mathcal{B}+3} + BB_{\mathcal{B}+1}$	$\cdot + \cdot$	–1	0	–1
$AB_{\mathcal{AB}+4} + AB_{\mathcal{BA}+4}$	$\cdot + \cdot$	–2	+1	–1

Proposition 9 *The recombinations with* $\Delta d = 0$ *involving cycles or circular singletons cannot create new components that can be used as sources of recombinations listed in Tables 1 and 2.*

The two sources of a recombination can also be called *partners.* Looking at Table 1 we observe that some types of paths have more partners than other types of paths. For example, all partners of $AB_{\mathcal{AB}+4}$ and $AB_{\mathcal{BA}+4}$ paths are also partners of $AA_{\mathcal{AB}+4}$ and $BB_{\mathcal{AB}+4}$ paths. Furthermore, some resultants of recombinations in Tables 1 and 2 can be used in other recombinations. These observations allow the identification of groups of recombinations, as listed in Table 3.

The deductions shown in Table 3 can be computed with an approach that greedily maximizes the number of recombinations in *U*, *V*, *W*, *X*, *Y* and *Z* in this order.

Table 2 Recombinations that have Δ*d* = 0 and create resultants that can be used in recombinations with Δ*d* ≤ –1 (listed in Table 1).

sources	resultants	$\Delta\sigma$	Δ_{dcj}	Δd
$AB_{\mathcal{AB}+4} + AB_{\mathcal{AB}+4}$	$AA_{\mathcal{A}+3} + BB_{\mathcal{B}+3}$	−2	+2	0
$AA_{\mathcal{A}+1} + AB_{\mathcal{BA}+4}q$	$\bullet + AA_{\mathcal{AB}+4}$	−1	+1	0
$AA_{\mathcal{B}+1} + AB_{\mathcal{AB}+4}$	$\bullet + AA_{\mathcal{AB}+4}q$	−1	+1	0
$BB_{\mathcal{A}+1} + AB_{\mathcal{AB}+4}$	$\bullet + BB_{\mathcal{AB}+4}$	−1	+1	0
$BB_{\mathcal{B}+1} + AB_{\mathcal{BA}+4}$	$\bullet + BB_{\mathcal{AB}+4}$	−1	+1	0
$AB_{\mathcal{BA}+4} + AB_{\mathcal{BA}+4}$	$AA_{\mathcal{B}+3} + BB_{\mathcal{A}+3}$	−2	+2	0
$AA_{\mathcal{AB}+2} + AB_{\mathcal{AB}+4}$	$\bullet + AA_{\mathcal{A}+1}$	−1	+1	0
$AA_{\mathcal{AB}+2} + AB_{\mathcal{BA}+4}$	$\bullet + AA_{\mathcal{B}+1}$	−1	+1	0
$BB_{\mathcal{AB}+2} + AB_{\mathcal{BA}+4}$	$\bullet + BB_{\mathcal{A}+1}$	−1	+1	0
$BB_{\mathcal{AB}+2} + AB_{\mathcal{AB}+4}$	$\bullet + BB_{\mathcal{B}+1}$	−1	+1	0

Table 3 All recombination groups obtained from Tables 1 and 2 (the recombinations from Table 2 appear only in groups in *Y* and *Z*). The column scr **indicates the contribution of each path in the distance decrease.**

	sources	resultants	Δd	scr
U	$AA_{\mathcal{AB}+4} + BB_{\mathcal{AB}+4}$	2•	−2	−1
V	$2AA_{\mathcal{AB}+4} + BB_{\mathcal{A}+1} + BB_{\mathcal{B}+1}$	4•	−3	−3/4
	$2BB_{\mathcal{AB}+4} + AA_{\mathcal{A}+1} + AA_{\mathcal{B}+1}$	4•	−3	−3/4
W	$AA_{\mathcal{AB}+4} + BB_{\mathcal{A}+1} + AB_{\mathcal{AB}+4}$	3•	−2	−2/3
	$AA_{\mathcal{AB}+4} + BB_{\mathcal{B}+1} + AB_{\mathcal{BA}+4}$	3•	−2	−2/3
	$BB_{\mathcal{AB}+4} + AA_{\mathcal{A}+1} + AB_{\mathcal{BA}+4}$	3•	−2	−2/3
	$BB_{\mathcal{AB}+4} + AA_{\mathcal{B}+1} + AB_{\mathcal{AB}+4}$	3•	−2	−2/3
	$2AA_{\mathcal{AB}+4} + BB_{\mathcal{A}+1}$	$2\bullet + AA_{\mathcal{B}+3}$	−2	−2/3
	$2AA_{\mathcal{AB}+4} + BB_{\mathcal{B}+1}$	$2\bullet + AA_{\mathcal{A}+3}$	−2	−2/3
	$2BB_{\mathcal{AB}+4} + AA_{\mathcal{A}+1}$	$2\bullet + BB_{\mathcal{B}+3}$	−2	−2/3
	$2BB_{\mathcal{AB}+4} + AA_{\mathcal{B}+1}$	$2\bullet + BB_{\mathcal{A}+3}$	−2	−2/3
X	Recombinations from Table 1 with *Δd* = −1		−1	−1/2
Y	$2AB_{\mathcal{AB}+4} + AA_{\mathcal{B}+1} + BB_{\mathcal{A}+1}$	4•	−2	−1/2
	$2AB_{\mathcal{BA}+4} + AA_{\mathcal{A}+1} + BB_{\mathcal{B}+1}$	4•	−2	−1/2
Z	$AB_{\mathcal{AB}+4} + AA_{\mathcal{AB}+2} + BB_{\mathcal{A}+3}$	3•	−1	−1/3
	$AB_{\mathcal{BA}+4} + AA_{\mathcal{AB}+2} + BB_{\mathcal{B}+3}$	3•	−1	−1/3
	$AB_{\mathcal{BA}+4} + AA_{\mathcal{A}+3} + BB_{\mathcal{AB}+2}$	3•	−1	−1/3
	$AB_{\mathcal{AB}+4} + AA_{\mathcal{B}+3} + BB_{\mathcal{AB}+2}$	3•	−1	−1/3
	$AB_{\mathcal{AB}+4} + AA_{\mathcal{B}+1} + BB_{\mathcal{A}+3}$	3•	−1	−1/3
	$AB_{\mathcal{AB}+4} + AA_{\mathcal{B}+3} + BB_{\mathcal{A}+1}$	3•	−1	−1/3
	$AB_{\mathcal{BA}+4} + AA_{\mathcal{A}+1} + BB_{\mathcal{B}+3}$	3•	−1	−1/3
	$AB_{\mathcal{BA}+4} + AA_{\mathcal{A}+3} + BB_{\mathcal{B}+1}$	3•	−1	−1/3
	$AB_{\mathcal{AB}+4} + AA_{\mathcal{B}+1} + BB_{\mathcal{A}+1}$	$2\bullet + AB_{\mathcal{BA}+4}$	−1	−1/3
	$AB_{\mathcal{BA}+4} + AA_{\mathcal{A}+1} + BB_{\mathcal{B}+1}$	$2\bullet + AB_{\mathcal{AB}+4}$	−1	−1/3
	$2AB_{\mathcal{AB}+4} + AA_{\mathcal{B}+1}$	$2\bullet + AA_{\mathcal{A}+3}$	−1	−1/3
	$2AB_{\mathcal{AB}+4} + BB_{\mathcal{A}+1}$	$2\bullet + BB_{\mathcal{B}+3}$	−1	−1/3
	$2AB_{\mathcal{BA}+4} + AA_{\mathcal{A}+1}$	$2\bullet + AA_{\mathcal{B}+3}$	−1	−1/3
	$2AB_{\mathcal{BA}+4} + BB_{\mathcal{B}+1}$	$2\bullet + BB_{\mathcal{A}+3}$	−1	−1/3

The *U* part contains only one operation and the two groups in V are mutually exclusive after applying U. The part *W* is then the application of all possible remaining groups of two operations with Δ*d* = −2. Similarly, the part X is only the application of all possible remaining operations with Δ*d* = −1. After X, the two groups in *Y* are mutually exclusive and then the same happens to the groups in Z. Although some groups in *W*, *X* and *Z* have some reusable resultants, those are actually never reused (if operations that are lower in the table use as sources resultants from higher operations, the sources of all referred operations would be previously consumed in operations that occupy even higher positions in the table). Due to this fact, the number of operations in *U*, *V* , *W*, *X*, *Y* and *Z* depends only on the initial number of each type of component.

With the results presented in this section we have an exact formula to compute the DCJ-substitution distance:

Theorem 3 *Given two genomes Aand Bwithout duplicated markers, we have*:

$$d^{sb}_{DCJ}(A,B) = d_{DCJ}(A,B) + \sum_{C \in AG(A,B)} \sigma(C) - P_L - P_C - 2U - 3V - 2W - X - 2Y - Z,$$

where P_L and P_C are the numbers of disjoint pairs of linear and circular singletons and U, V, W, X, Y and Z are computed as described above.

The formula given in Theorem 3 is analogous to the one which we have obtained in [6] to compute the DCJ-indel distance. Both formulas depend on factors that can be computed in linear time [6].

Triangular inequality

Note that, since only unique markers can be substituted in this model, we avoid the “free lunch problem”, mentioned in [5], that is the possibility of transforming any genome A into any genome *B* by simply substituting the whole content of A by the whole content of B. However, the triangular inequality can be disrupted in the DCJ-substitution distance. In other words, given any three genomes *A*, *B* and *C* without duplicated markers, there is no guarantee that the triangular inequality $d^{sb}_{DCJ}(A,B) \leq d^{sb}_{DCJ}(A,c) + d^{sb}_{DCJ}(B,C)$ holds. In a companion paper [11] we provide an efficient way to establish the triangular inequality *a posteriori* in both the DCJ-indel [6] and the DCJ-substitution distances.

Experiments

The objective of this model is to provide a parsimonious genomic distance, that in practice is a lower bound to real distances. Nevertheless, we could run some experiments on data from human *X* and *Y* chromosomes and obtained a parsimonious sorting scenario that is coherent with the results available in the literature. During evolution, a portion of the human *Y* chromosome has become increasingly subjected to local mutations, while the X chromosome remained relatively conserved, as we will see in the following. Human X and Y chromosomes are very different and, while X is 155 Mbp long, the Y chromosome is 58 Mbp long. However, they still share *pseudo-autosomal* regions at both extremities and are believed to have evolved from an identical autosomal pair [12] (the autosomes are all non-sex chromosomes). Current theories suggest that the pseudo-autosomal region, which originally covered the whole chromosomes, was successively pruned by a few big inversions on the Y chromosome [13] (we call these inversions *pruning*). After each pruning inversion, several mutations seem to have occurred on the affected part of the Y chromosome, while X remained "closer" to the common ancestor.

A parsimonious scenario of 8 inversions on the markers common to chromosomes X and Y has been published in [[10], Fig. 7], and is given as an argument to support the existence and bounds of the three most recent pruning inversions, but unique markers were simply ignored. We used our method to compute the DCJ-substitution distance using the same dataset, but reincorporating the unique markers, and obtained a DCJ-substitution distance of 14. Then we manually reconstructed the evolutionary scenario of human chromosomes X and Y and obtained a parsimonious scenario with 8 inversions and 6 substitutions (including 2 insertions and 1 deletion) that is coherent with the pruning inversions given in [10] (see Figure 5). Although a DCJ is a very comprehensive operation and can represent many rearrangement events, in the analysis of unichromosomal genomes DCJs often represent only inversions, and this also happens in this dataset.

Discussion

Our method was designed to find gene mutations, but it could also help to improve orthology assignments, that are the computational prediction of orthologous pairs of genes from different species. No orthology predictor is able to find all assignments correctly. In particular, when comparing two different species, some pairs of orthologous genes that are below the predictor threshold remain unassigned. Since our substitutions establish a relation between different genes in the two compared genomes, they correspond to candidates to be assigned as orthologous genes.

Conclusions and future work

In this work we presented a new model to compare two genomes with unequal content, but without duplicated markers, using substitutions and DCJ operations, and developed a linear time algorithm to exactly compute the DCJ-substitution distance.

Although the objective of this model is to provide a parsimonious genomic distance, that in practice is a lower bound to real distances, based on our method we have manually reconstructed a parsimonious evolutionary scenario of human chromosomes X and Y. We considered biological constraints that are specific to this case and obtained a scenario that is coherent with the results given in the literature.

By reconstructing a parsimonious scenario that minimizes substitutions, we may identify genomic regions that were subject to continuous mutations during evolution and could have a common evolutionary origin.

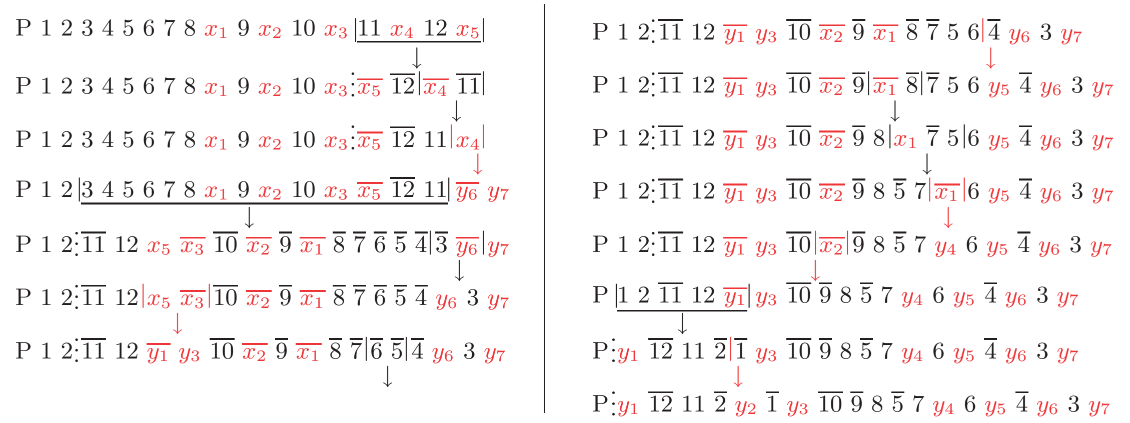
$P\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ x_1\ 9\ x_2\ 10\ x_3\ |\underline{11\ x_4\ 12\ x_5}|$
$P\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ x_1\ 9\ x_2\ 10\ x_3 \vdots \overline{x_5}\ \overline{12}|\overline{x_4}\ \overline{11}|$
$P\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ x_1\ 9\ x_2\ 10\ x_3 \vdots \overline{x_5}\ \overline{12}\ 11|x_4|$
$P\ 1\ 2|\underline{3\ 4\ 5\ 6\ 7\ 8\ x_1\ 9\ x_2\ 10\ x_3\ \overline{x_5}\ \overline{12}\ 11}|\ \overline{y_6}\ y_7$
$P\ 1\ 2 \vdots \overline{11}\ 12\ x_5\ \overline{x_3}\ \overline{10}\ \overline{x_2}\ \overline{9}\ \overline{x_1}\ \overline{8}\ \overline{7}\ \overline{6}\ \overline{5}\ \overline{4}|\overline{3}\ \overline{y_6}|y_7$
$P\ 1\ 2 \vdots \overline{11}\ 12|x_5\ \overline{x_3}|\overline{10}\ \overline{x_2}\ \overline{9}\ \overline{x_1}\ \overline{8}\ \overline{7}\ \overline{6}\ \overline{5}\ \overline{4}\ y_6\ 3\ y_7$
$P\ 1\ 2 \vdots \overline{11}\ 12\ \overline{y_1}\ y_3\ \overline{10}\ \overline{x_2}\ \overline{9}\ \overline{x_1}\ \overline{8}\ \overline{7}|\overline{6}\ \overline{5}|\overline{4}\ y_6\ 3\ y_7$
$P\ 1\ 2 \vdots \overline{11}\ 12\ \overline{y_1}\ y_3\ \overline{10}\ \overline{x_2}\ \overline{9}\ \overline{x_1}\ \overline{8}\ \overline{7}\ 5\ 6|\overline{4}\ y_6\ 3\ y_7$
$P\ 1\ 2 \vdots \overline{11}\ 12\ \overline{y_1}\ y_3\ \overline{10}\ \overline{x_2}\ \overline{9}|\overline{x_1}\ \overline{8}|\overline{7}\ 5\ 6\ y_5\ \overline{4}\ y_6\ 3\ y_7$
$P\ 1\ 2 \vdots \overline{11}\ 12\ \overline{y_1}\ y_3\ \overline{10}\ \overline{x_2}\ \overline{9}\ 8|x_1\ \overline{7}\ 5|6\ y_5\ \overline{4}\ y_6\ 3\ y_7$
$P\ 1\ 2 \vdots \overline{11}\ 12\ \overline{y_1}\ y_3\ \overline{10}\ \overline{x_2}\ \overline{9}\ 8\ \overline{5}\ 7|\overline{x_1}|6\ y_5\ \overline{4}\ y_6\ 3\ y_7$
$P\ 1\ 2 \vdots \overline{11}\ 12\ \overline{y_1}\ y_3\ \overline{10}|\overline{x_2}|\overline{9}\ 8\ \overline{5}\ 7\ y_4\ 6\ y_5\ \overline{4}\ y_6\ 3\ y_7$
$P\ |\underline{1\ 2\ \overline{11}\ 12\ \overline{y_1}}|\ y_3\ \overline{10}\ \overline{9}\ 8\ \overline{5}\ 7\ y_4\ 6\ y_5\ \overline{4}\ y_6\ 3\ y_7$
$P \vdots y_1\ \overline{12}\ 11\ \overline{2}|\overline{1}\ y_3\ \overline{10}\ \overline{9}\ 8\ \overline{5}\ 7\ y_4\ 6\ y_5\ \overline{4}\ y_6\ 3\ y_7$
$P \vdots y_1\ \overline{12}\ 11\ \overline{2}\ y_2\ \overline{1}\ y_3\ \overline{10}\ \overline{9}\ 8\ \overline{5}\ 7\ y_4\ 6\ y_5\ \overline{4}\ y_6\ 3\ y_7$

Figure 5 A parsimonious scenario of 8 inversions and 6 substitutions (including 2 insertions and 1 deletion) sorting human X into Y chromosome, using the dataset given in [10]. The symbol 'P' represents the current pseudo-autosomal region in the beginning of X and Y. Each number represents a common marker, each symbol x_i represents a unique marker in X and each symbol y_i represents a unique marker in Y (the unique markers were also obtained from the data in [10]). The three pruning inversions suggested in [[10], Fig. 7] are underlined. The boundary of the pseudo-autosomal region, indicated with vertical dots, is shifted to the left after each pruning inversion.

Currently our method is only able to compute the genomic distance, but in a future work we intend to study the space of all parsimonious sorting scenarios and develop methods to systematically identify such regions.

The DCJ-substitution model could also be used to refine orthology assignments, since in some cases a substitution could actually correspond to an unannotated orthology. We also plan on exploring the use of our method in refining orthology in a future work.

Acknowledgements
This research was partially supported by the Brazilian research agencies CNPq (grant PROMETRO 563087/2010-2) and FAPERJ (grant INST E-26/111.837/2010).
This article has been published as part of *BMC Bioinformatics* Volume 12 Supplement 9, 2011: Proceedings of the Ninth Annual Research in Computational Molecular Biology (RECOMB) Satellite Workshop on Comparative Genomics. The full contents of the supplement are available online at http://www.biomedcentral.com/1471-2105/12?issue=S9.

Author details
[1]Instituto Nacional de Metrologia, Qualidade e Tecnologia, Duque de Caxias, 25250-020, Brazil. [2]AG Genominformatik, Technische Fakultät, Universität Bielefeld, Bielefeld, 33594, Germany.

Authors' contributions
MDVB and JS have elaborated the model. MDVB, RM, LCR and JS have proved the results and written the paper. MDVB has also run the experiments.

Competing interests
The authors declare that they have no competing interests.

Published: 5 October 2011

References

1. Bergeron A, Mixtacki J, Stoye J: **A unifying view of genome rearrangements.** *Proc. of WABI 2006, LNBI* 2006, **4175**:163-173.
2. Braga MDV, Stoye J: **The solution space of sorting by DCJ.** *Journal of Computational Biology* 2010, **17(9)**:1145-1165.
3. Hannenhalli S, Pevzner P: **Transforming men into mice (polynomial algorithm for genomic distance problem).** *Proc. of FOCS* 1995, 581-592.
4. El-Mabrouk N: **Sorting Signed Permutations by Reversals and Insertions/Deletions of Contiguous Segments.** *Journal of Discrete Algorithms* 2001, **1**:105-122.
5. Yancopoulos S, Friedberg R: **DCJ path formulation for genome transformations which include insertions, deletions, and duplications.** *Journal of Computational Biology* 2009, **16(10)**:1311-1338.
6. Braga MDV, Willing E, Stoye J: **Double Cut and Join with Insertions and Deletions.** *Journal of Computational Biology* 2011, **18**:1167-1184, DOI: 10.1089/cmb.2011.0118.
7. Yancopoulos S, Attie O, Friedberg R: **Efficient sorting of genomic permutations by translocation, inversion and block interchange.** *Bioinformatics* 2005, **21**:3340-3346.
8. Boore JL: **The duplication/random loss model for gene rearrangement exemplified by mitochondrial genomes of deuterostome animals.** In *Comparative Genomics* Sankoff D, Nadeau JH 2000, 133-148.
9. Moritz C, Dowling TE, Brown WM: **Evolution of animal mitochondrial DNA: relevance for population biology and systematics.** *Annu. Rev. Ecol. Syst* 1987, **18**:269-292.
10. Ross MT, *et al*: **The DNA sequence of the human X chromosome.** *Nature* 2005, **434**:325-337.
11. Braga MDV, Machado R, Ribeiro LC, Stoye J: **On the weight of indels in genomic distances.** *BMC Bioinformatics* 2011, **12(Suppl 9)**:S13, doi:10.1186/1471-2105-12-S9-S13.
12. Ohno S: **Sex chromosomes and sex-linked genes.** Springer-Verlag, Berlin; 1967.
13. Lahn BT, Page DC: **Four evolutionary strata on the human X chromosome.** *Science* 1999, **286**:964-967.

doi:10.1186/1471-2105-12-S9-S8
Cite this article as: Braga *et al.*: Genomic distance under gene substitutions. *BMC Bioinformatics* 2011 **12**(Suppl 9):S8.