Martinez *et al. Algorithms for Molecular Biology* (2015) 10:13
DOI 10.1186/s13015-015-0041-9

ALGORITHMS FOR MOLECULAR BIOLOGY

RESEARCH Open Access

On the family-free DCJ distance and similarity

Fábio V Martinez[1,2], Pedro Feijão[2], Marília DV Braga[3] and Jens Stoye[2*]

Abstract

Structural variation in genomes can be revealed by many (dis)similarity measures. Rearrangement operations, such as the so called double-cut-and-join (DCJ), are large-scale mutations that can create complex changes and produce such variations in genomes. A basic task in comparative genomics is to find the rearrangement distance between two given genomes, i.e., the minimum number of rearragement operations that transform one given genome into another one. In a family-based setting, genes are grouped into gene families and efficient algorithms have already been presented to compute the DCJ distance between two given genomes. In this work we propose the problem of computing the DCJ distance of two given genomes without prior gene family assignment, directly using the pairwise similarities between genes. We prove that this new family-free DCJ distance problem is APX-hard and provide an integer linear program to its solution. We also study a family-free DCJ similarity and prove that its computation is NP-hard.

Keywords: Genome rearrangement, DCJ, Family-free genome comparison

Background

Genomes are subject to mutations or rearrangements in the course of evolution. Typical large-scale rearrangements change the number of chromosomes and/or the positions and orientations of genes. Examples of such rearrangements are inversions, translocations, fusions and fissions. A classical problem in comparative genomics is to compute the rearrangement distance, that is, the minimum number of rearrangements required to transform a given genome into another given genome [1].

In order to study this problem, one usually adopts a high-level view of genomes, in which only "relevant" fragments of the DNA (e.g., genes) are taken into consideration. Furthermore, a pre-processing of the data is required, so that we can compare the content of the genomes.

One popular method, adopted for more than 20 years, is to group the genes in both genomes into *gene families*, so that two genes in the same family are said to be equivalent. This setting is said to be *family-based*. Without gene duplications, that is, with the additional restriction that each family occurs exactly once in each genome, many polynomial models have been proposed to compute the genomic distance [2-5]. However, when gene duplications are allowed, the problem is more intrincate and all approaches proposed so far are NP-hard, see for instance [6-10].

It is not always possible to classify each gene unambiguously into a single gene family. Due to this fact, an alternative to the family-based setting was proposed recently and consists in studying the rearrangement distance without prior family assignment. Instead of families, the pairwise similarity between genes is directly used [11,12]. This approach is said to be *family-free*. Although the family-free setting seems to be at least as difficult as the family-based setting with duplications, its complexity is still unknown for various distance models.

In this work we are interested in the problem of computing the distance of two given genomes in a family-free setting, using the *double cut and join* (DCJ) model [5]. The DCJ operation, that consists of cutting a genome in two distinct positions and joining the four resultant open ends in a different way, represents most of large-scale rearrangements that modify genomes. After preliminaries and a formal definition of the family-free DCJ distance, we present a hardness result, before giving a linear programming solution and showing its feasibility for practical problem instances. Finally, we also study the problem of computing the similarity – a counterpart of the distance function – of two given genomes in a family-free setting using the DCJ model and show its NP-hardness.

*Correspondence: jens.stoye@uni-bielefeld.de
[2]Technische Fakultät and CeBiTec, Universität Bielefeld, Universitätsstr. 25, 33615 Bielefeld, Germany
Full list of author information is available at the end of the article

This paper is an extended version of [13], that was presented at the 14th Workshop on Algorithms in Bioinformatics, WABI 2014.

Preliminaries

Each gene g in a genome is an oriented DNA fragment that can be represented by the symbol g itself, if it has direct orientation, or by the symbol $-g$, if it has reverse orientation. Furthermore, each one of the two extremities of a linear chromosome is called a *telomere*, represented by the symbol $\circ$. Each chromosome in a genome can be represented by a string that can be circular, if the chromosome is circular, or linear and flanked by the symbols $\circ$ if the chromosome is linear. For the sake of clarity, each chromosome is also flanked by parentheses. As an example, consider the genome $A = \{(\circ\ 3\ {-1}\ 4\ 2\ \circ), (\circ\ 5\ {-6}\ {-7}\ \circ)\}$ that is composed of two linear chromosomes.

Since a gene g has an orientation, we can distinguish its two ends, also called its *extremities*, and denote them by g^t (*tail*) and g^h (*head*). An *adjacency* in a genome is either the extremity of a gene that is adjacent to one of its telomeres, or a pair of consecutive gene extremities in one of its chromosomes. If we consider again the genome A above, the adjacencies in its first chromosome are 3^t, 3^h1^h, 1^t4^t, 4^h2^t and 2^h.

Throughout this paper, let A and B be two distinct genomes and let $\mathcal{A}$ be the set of genes in genome A and $\mathcal{B}$ be the set of genes in genome B.

Adjacency graph and family-based DCJ distance

In the family-based setting the two genomes A and B have the same content, that is, $\mathcal{A} = \mathcal{B}$. When there are no duplications, that is, when each family is represented by exactly one gene in each genome, the DCJ distance can be easily computed with the help of the *adjacency graph* $AG(A, B)$, a bipartite multigraph such that each partition corresponds to the set of adjacencies of one of the two input genomes and an edge connects the same extremities of genes in both genomes. In other words, there is a one-to-one correspondence between the set of edges in $AG(A, B)$ and the set of gene extremities. Vertices have degree one or two and thus an adjacency graph is a collection of paths and cycles. An example of an adjacency graph is given in Figure 1.

The family-based DCJ distance $\mathrm{d_{DCJ}}$ between two genomes A and B without duplications can be computed in linear time and is closely related to the number of components in the adjacency graph $AG(A, B)$ [2]:

$$\mathrm{d_{DCJ}}(A, B) = n - c - i/2\,,$$

where $n = |\mathcal{A}| = |\mathcal{B}|$ is the number of genes in both genomes, c is the number of cycles and i is the number of odd paths in $AG(A, B)$.

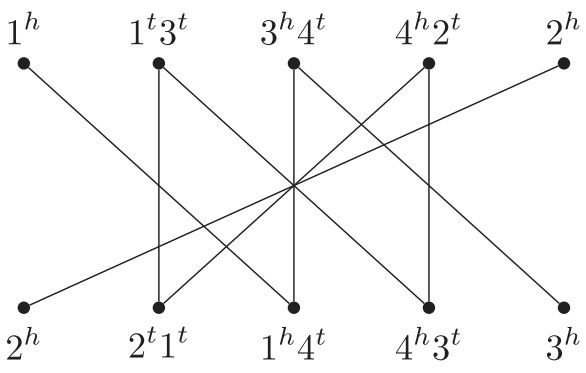

Figure 1 The adjacency graph for the two unichromosomal and linear genomes $A = \{(\circ\ {-1}\ 3\ 4\ 2\ \circ)\}$ and $B = \{(\circ\ {-2}\ 1\ 4\ 3\ \circ)\}$.

Observe that, in Figure 1, the number of genes is $n = 4$ and $AG(A, B)$ has one cycle and two odd paths. Consequently the DCJ distance is $\mathrm{d_{DCJ}}(A, B) = 4 - 1 - 2/2 = 2$.

The formula for $\mathrm{d_{DCJ}}(A, B)$ can also be derived using the following approach. Given a component C in $AG(A, B)$, let $|C|$ denote the length, or number of edges, of C. From [14,15] we know that each component in $AG(A, B)$ contributes independently to the DCJ distance, depending uniquely on its length. Formally, the contribution $d(C)$ of a component C in the total distance is given by:

$$d(C) = \begin{cases} \frac{|C|}{2} - 1\,, & \text{if } C \text{ is a cycle}\,, \\ \frac{|C|-1}{2}\,, & \text{if } C \text{ is an odd path}\,, \\ \frac{|C|}{2}\,, & \text{if } C \text{ is an even path}\,. \end{cases}$$

The sum of the lengths of all components in the adjacency graph is equal to $2n$. Let $\mathcal{C}$, $\mathcal{I}$, and $\mathcal{P}$ represent the sets of components in $AG(A, B)$ that are cycles, odd paths and even paths, respectively. Then, the DCJ distance can be calculated as the sum of the contributions of each component:

$$\begin{aligned} \mathrm{d_{DCJ}}(A, B) &= \sum_{C \in AG(A,B)} d(C) \\ &= \sum_{C \in \mathcal{C}} \left(\frac{|C|}{2} - 1\right) + \sum_{C \in \mathcal{I}} \left(\frac{|C| - 1}{2}\right) + \sum_{C \in \mathcal{P}} \left(\frac{|C|}{2}\right) \\ &= \frac{1}{2}\left(\sum_{C \in AG(A,B)} |C|\right) - \sum_{C \in \mathcal{C}} 1 - \sum_{C \in \mathcal{I}} \frac{1}{2} \\ &= n - c - i/2\,. \end{aligned}$$

Gene similarity graph for the family-free model

In the family-free setting, each gene in each genome is represented by a distinct symbol, thus $\mathcal{A} \cap \mathcal{B} = \emptyset$ and the

cardinalities $|\mathcal{A}|$ and $|\mathcal{B}|$ may be distinct. Let a be a gene in A and b be a gene in B, then their *normalized similarity* is given by the value $\sigma(a,b)$ that ranges in the interval $[0,1]$.

We can represent the similarities between the genes of genome A and the genes of genome B with respect to σ in the so called *gene similarity graph* [12], denoted by $GS_\sigma(A,B)$. This is a weighted bipartite graph whose partitions $\mathcal{A}$ and $\mathcal{B}$ are the sets of genes in genomes A and B, respectively. Furthermore, for each pair of genes (a,b), such that $a \in \mathcal{A}$ and $b \in \mathcal{B}$, if $\sigma(a,b) > 0$ there is an edge e connecting a and b in $GS_\sigma(A,B)$ whose weight is $\sigma(e) := \sigma(a,b)$. An example of a gene similarity graph is given in Figure 2.

Reduced genomes and their weighted adjacency graph

Let A and B be two genomes and let $GS_\sigma(A,B)$ be their gene similarity graph. Now let $M = \{e_1, e_2, \ldots, e_n\}$ be a matching in $GS_\sigma(A,B)$ and denote by $w(M) = \sum_{e_i \in M} \sigma(e_i)$ the weight of M, that is the sum of its edge weights. Since the endpoints of each edge $e_i = (a,b)$ in M are not saturated by any other edge of M, we can unambiguously define the function $\ell^M(a) = \ell^M(b) = i$. The *reduced genome* A^M is obtained by deleting from A all genes that are not saturated by M, and renaming each saturated gene a to $\ell^M(a)$, preserving its orientation. Similarly, the reduced genome B^M is obtained by deleting from B all genes that are not saturated by M, and renaming each saturated gene b to $\ell^M(b)$, preserving its orientation. Observe that the set of genes in A^M and in B^M is $\mathcal{G}(M) = \{\ell^M(g) : g \text{ is saturated by the matching } M\} = \{1, 2, \ldots, n\}$.

Let A^M and B^M be the reduced genomes for a given matching M of $GS_\sigma(A,B)$. The *weighted adjacency graph* of A^M and B^M, denoted by $AG_\sigma(A^M, B^M)$, is obtained by constructing the adjacency graph of A^M and B^M and adding weights to the edges as follows. For each gene i in $\mathcal{G}(M)$, both edges $i^t i^t$ and $i^h i^h$ inherit the weight of edge e_i in M, that is, $\sigma(i^t i^t) = \sigma(i^h i^h) = \sigma(e_i)$. Observe that, for each edge $e \in M$, we have two edges of weight $\sigma(e)$ in $AG_\sigma(A^M, B^M)$, thus $w(AG_\sigma(A^M, B^M)) = 2\,w(M)$ (the weight of $AG_\sigma(A^M, B^M)$ is twice the weight of M). Examples of weighted adjacency graphs are shown in Figure 3.

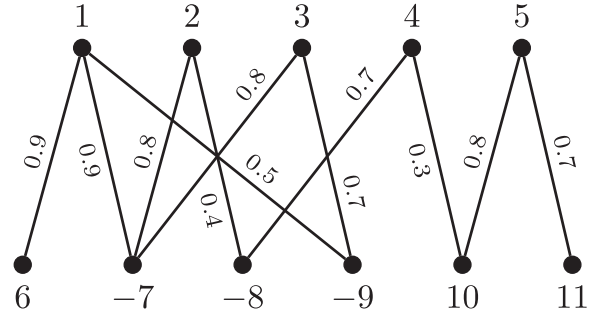

Figure 2 A possible gene similarity graph for the two unichromosomal linear genomes $A = \{(\circ\ 1\ 2\ 3\ 4\ 5\ \circ)\}$ and $B = \{(\circ\ 6\ -7\ -8\ -9\ 10\ 11\ \circ)\}$.

The family-free DCJ distance

Based on the weighted adjacency graph, in [12] a family-free DCJ *similarity* measure has been proposed. We will come back to this measure later in this paper. Before that, to be more consistent with the comparative genomics literature, where distance measures are more common than similarities, here we also propose a family-free DCJ *distance*. This family-free distance is based on the weighted DCJ distance of reduced genomes. An important design criterion for this definition is that it must be the same as the (unweighted) family-based DCJ distance when all weights are equal to 1.

The first step in our definition is to consider the components of the graph $AG_\sigma(A^M, B^M)$ separately, similarly to the approach described previously for the family-based model. Here the contribution of each component C is denoted by $d_\sigma(C)$ and must include not only the length $|C|$ of the component, but also information about the weights of the edges in C. Basically, we need a function $f(C)$ to use instead of $|C|$ in the contribution function $d_\sigma(C)$, such that: (*i*) when all edges in C have weight 1, $f(C) = |C|$, that is, the contribution of C is the same as in the family-based version; (*ii*) when the weights decrease, f should increase, because smaller weights mean less similarity, or increased distance between the genomes.

The simplest linear function f that satisfies both conditions is $f(C) = 2|C| - w(C)$, where $w(C) = \sum_{e \in C} \sigma(e)$ is the sum of the weights of all the edges in C. Then, the *weighted contribution* $d_\sigma(C)$ of the different types of components is:

$$d_\sigma(C) = \begin{cases} \frac{2|C| - w(C)}{2} - 1\,, & \text{if } C \text{ is a cycle}\,, \\ \frac{2|C| - w(C) - 1}{2}\,, & \text{if } C \text{ is an odd path}\,, \\ \frac{2|C| - w(C)}{2}\,, & \text{if } C \text{ is an even path}\,. \end{cases}$$

Let $\mathcal{C}$, $\mathcal{I}$, and $\mathcal{P}$ represent the sets of components in $AG_\sigma(A^M, B^M)$ that are cycles, odd paths and even paths, respectively. Summing the contributions of all the

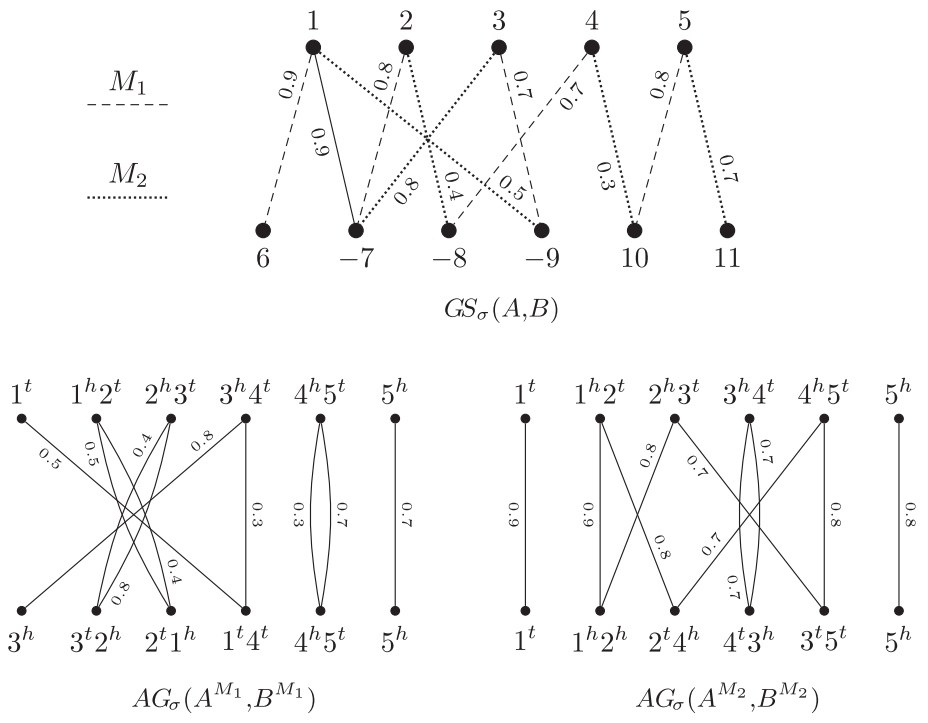

Figure 3 Reduced genomes and their weighted adjacency graph. Considering the genomes $A = \{(\circ\ 1\ 2\ 3\ 4\ 5\ \circ)\}$ and $B = \{(\circ\ 6\ -7\ -8\ -9\ 10\ 11\ \circ)\}$ as in Figure 2, let M_1 (dotted edges) and M_2 (dashed edges) be two distinct matchings in $GS_\sigma(A,B)$, shown in the upper part. The two resulting weighted adjacency graphs $AG_\sigma(A^{M_1},B^{M_1})$, that has two odd paths and three cycles, and $AG_\sigma(A^{M_2},B^{M_2})$, that has two odd paths and two cycles, are shown in the lower part.

components, the resulting distance for a certain matching M is computed as follows:

$$\begin{aligned} d_\sigma(A^M,B^M) &= \sum_{C\in AG_\sigma(A^M,B^M)} d_\sigma(C) \\ &= \sum_{C\in\mathcal{C}}\left(\frac{2|C|-w(C)}{2}-1\right)+\sum_{C\in\mathcal{I}}\left(\frac{2|C|-w(C)-1}{2}\right) \\ &\quad + \sum_{C\in\mathcal{P}}\left(\frac{2|C|-w(C)}{2}\right) \\ &= \sum_{C\in AG_\sigma(A^M,B^M)} |C| - \frac{1}{2}\left(\sum_{C\in AG_\sigma(A^M,B^M)} w(C)\right) \\ &\quad - \sum_{C\in\mathcal{C}} 1 - \sum_{C\in\mathcal{I}}\frac{1}{2} \\ &= 2|M| - w(AG_\sigma(A^M,B^M))/2 - c - i/2 \\ &= \mathrm{d}_{\mathrm{DCJ}}(A^M,B^M) + |M| - w(M)\,, \end{aligned} \tag{1}$$

since the number of genes in $\mathcal{G}(M)$ is equal to the size of M.

Observe that not only the components of the graph, but also the size and the weight of the matching influence the distance above. For example, in Figure 3, matching M_1 gives the weighted adjacency graph with more components, but whose distance $d_\sigma(A^{M_1},B^{M_1}) = 1 + 5 - 2.7 = 3.3$ is larger. On the other hand, M_2 gives the weighted adjacency graph with less components, but whose distance $d_\sigma(A^{M_2},B^{M_2}) = 2 + 5 - 3.9 = 3.1$ is smaller.

Our goal in the following sections is to study the problem of computing the family-free DCJ distance, i.e., to find a matching in $GS_\sigma(A,B)$ that minimizes d_σ. First of all, it is important to observe that the behaviour of this function does not correlate with the size of the matching. Often smaller matchings, that possibly discard gene assignments, lead to smaller distances. Actually, it is easy to see that, for any pair of genomes with any gene similarity graph, a trivial empty matching leads to the minimum distance, equal to zero. Due to this fact we restrict the distance to *maximal matchings* only. This ensures that no pairs of genes with positive similarity score are simply discarded, even though they might increase the overall distance. Hence we have the following optimization problem:

Problem FFDCJ-DISTANCE(A,B): Given genomes A and B and their gene similarities σ, calculate their family-free DCJ distance

$$\mathrm{d}_{\mathrm{FFDCJ}}(A,B) = \min_{M\in\mathbb{M}}\{d_\sigma(A^M,B^M)\}\,, \tag{2}$$

where $\mathbb{M}$ is the set of all maximal matchings in $GS_\sigma(A,B)$.

Complexity of the family-free DCJ distance

In order to assess the complexity of FFDCJ-DISTANCE, we use a restricted version of the family-based *exemplar DCJ distance problem* [6,8]:

Problem (s,t)-EXDCJ-DISTANCE(A,B): Given genomes A and B, where each family occurs at most s times in A and at most t times in B, obtain *exemplar* genomes A' and B' by removing all but one copy of each family in each genome, so that the DCJ distance $d_{DCJ}(A',B')$ is minimized.

We establish the computational complexity of the FFDCJ-DISTANCE problem by means of a polynomial time and approximation preserving (AP-) reduction from the problem $(1,2)$-EXDCJ-DISTANCE, which is NP-hard [8]. Note that the authors of [8] only consider unichromosomal genomes, but the reduction can be extended to multichromosomal genomes, since an algorithm that solves the multichromosomal case also solves the unichromosomal case.

Theorem 1. *Problem* FFDCJ-DISTANCE*(A,B) is APX-hard, even if the maximum degrees in the two partitions of $GS_\sigma(A,B)$ are respectively one and two.*

Before proving the result, we need some definitions and particularly a formal definition of an AP-reduction. These definitions are based on [16].

An *optimization problem* is defined by three main elements: a set of instances, a set $\mathrm{Sol}(I)$ of *feasible solutions* for each instance I, and a function val that relates a non-negative rational number $\mathrm{val}(I,S)$ to each instance I and solution S in $\mathrm{Sol}(I)$. Thus, in a minimization problem, the aim is to find a feasible solution of minimum value. That is, if Π is an optimization problem with an instance I, then we want to find $S \in \mathrm{Sol}(I)$ that minimizes $\mathrm{val}(I,S)$, called an *optimal solution* to the optimization problem. For an instance I, the value of an optimal solution is denoted by $\mathrm{opt}(I)$.

An *AP-reduction* from an optimization problem Π to an optimization problem Π' is a triple (f,g,β), where f and g are algorithms and β is a positive rational number, such that:

(AP1) f receives as input a positive rational number δ and an instance I of Π, and returns an instance $f(\delta,I)$ of Π';

(AP2) g receives as input a positive rational number δ, an instance I of Π and an element S' in $\mathrm{Sol}(f(\delta,I))$, and returns a solution $g(\delta,I,S')$ in $\mathrm{Sol}(I)$;

(AP3) for any positive rational number δ, $f(\delta,\cdot)$ and $g(\delta,\cdot,\cdot)$ are polynomial time algorithms;

(AP4) for any instance I of Π, any positive rational number δ, and any S' in $\mathrm{Sol}(f(\delta,I))$, if

$$\mathrm{val}(f(\delta,I),S') \leq (1+\delta)\,\mathrm{opt}(f(\delta,I))\,,$$

then

$$\mathrm{val}(I,g(\delta,I,S')) \leq (1+\beta\delta)\,\mathrm{opt}(I)\,.$$

An AP-reduction from Π to Π' is frequently denoted by $\Pi \leq_{\mathrm{AP}} \Pi'$, and we say that Π is *AP-reduced* to Π'. An AP-reduction is a special type of reduction which preserves both the polynomiality property and the approximation factor.

Now, we can proceed with the proof of Theorem 1.

Proof (of Theorem 1). We give an AP-reduction (f,g,β) from $(1,2)$-EXDCJ-DISTANCE to FFDCJ-DISTANCE.

(AP1) Algorithm f receives as input a positive rational number δ and an instance (A,B) of $(1,2)$-EXDCJ-DISTANCE where A and B are genomes from a set of genes $\mathcal{G}$ and each gene in $\mathcal{G}$ occurs at most once in A and at most twice in B, and constructs an instance $(A_\mathrm{F},B_\mathrm{F}) = f(\delta,(A,B))$ of FFDCJ-DISTANCE as follows. Let the genes of A be denoted $a_1,a_2,\ldots,a_{|A|}$ and the genes of B be denoted $b_1,b_2,\ldots,b_{|B|}$. Then A_F and B_F are copies of A and B, respectively, except that symbol a_i in A_F is relabeled by i, keeping its orientation, and b_j in B_F is relabeled by $j+|A|$, also keeping its orientation. Furthermore, the normalized similarity measure σ for genes in A_F and B_F is defined as $\sigma(i,k)=1$ for i in A_F and k in B_F, such that a_i is in A, b_j is in B, a_i and b_j are in the same gene family, and $k=j+|A|$. Otherwise, $\sigma(i,k)=0$. Note that the construction is independent of the value of δ. Figure 4 refers to an example of a gene similarity graph $GS_\sigma(A_\mathrm{F},B_\mathrm{F})$ of this construction.

(AP2) Algorithm g receives as input a positive rational number δ, an instance (A,B) of $(1,2)$-EXDCJ-DISTANCE and a solution M of FFDCJ-DISTANCE, and transforms M

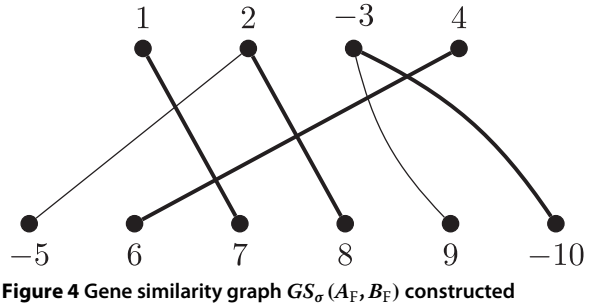

Figure 4 Gene similarity graph $GS_\sigma(A_\mathrm{F},B_\mathrm{F})$ constructed from the input genomes $A=\{(\circ\ a\ c\ -b\ d\ \circ)\}$ and $B=\{(\circ\ -c\ d\ a\ c\ b\ -b\ \circ)\}$ of $(1,2)$-EXDCJ-DISTANCE, where all edge weights are 1. Highlighted edges represent a maximal matching M in $GS_\sigma(A_\mathrm{F},B_\mathrm{F})$.

into a solution (A_X, B_X) of $(1,2)$-EXDCJ-DISTANCE. This is a simple construction: for each edge (i,k) in M, we add symbols a_i to A_X and b_j to B_X, where $j = k - |A|$. The value of δ does not influence the construction. In the example of Figure 4, a matching $M = \{(1,7),(2,8),(-3,-10),(4,6)\}$, which is a solution to FFDCJ-DISTANCE(A_F, B_F), is transformed by g into the genomes $A_X = \{(\circ\ a_1\ a_2\ a_3\ a_4\ \circ)\} = \{(\circ\ a\ \ c\ \ -b\ \ d\ \circ)\}$ and $B_X = \{(\circ\ b_2\ \ b_3\ \ b_4\ \ b_6\ \circ)\} = \{(\circ\ d\ \ a\ \ c\ \ -b\ \circ)\}$, which is a solution to $(1,2)$-EXDCJ-DISTANCE(A,B).

(AP3) Clearly, for any positive rational number δ, functions f and g are polynomial time algorithms on the size of their respective instances. A schematic view of these transformations is presented below.

$$\begin{array}{ccc} (1,2)\text{-EXDCJ-DISTANCE} & & \text{FFDCJ-DISTANCE} \\ (A,B) & \xrightarrow{f,\delta} & (A_F,B_F) := f(\delta,(A,B)) \\ \downarrow & & \downarrow \\ (A_X,B_X) := g(\delta,(A,B),M) & \xleftarrow{g,\delta} & M \end{array}$$

(AP4) Finally, suppose that for an instance (A,B) of $(1,2)$-EXDCJ-DISTANCE, a positive rational number δ and a solution M of FFDCJ-DISTANCE with instance $(A_F,B_F) = f(\delta,(A,B))$, we have

$$d_\sigma(A_F^M, B_F^M) \leq (1+\delta)\,\mathrm{opt}(\text{FFDCJ-DISTANCE}(A_F,B_F))\,.$$

Let $A_X := A$ and B_X be an exemplar genome of B, such that $(A_X,B_X) = g(\delta,(A,B),M)$. We want to prove that (A_X,B_X) is such that

$$d(A_X,B_X) \leq (1+\beta\delta)\,\mathrm{opt}((1,2)\text{-EXDCJ-DISTANCE}(A,B)) \tag{3}$$

for some fixed positive rational number β.

Denote by c_{AG} and i_{AG} the number of cycles and odd paths, respectively, in the adjacency graph $AG(A_X,B_X)$, and by c_{AG_σ} and i_{AG_σ} the number of cycles and odd paths, respectively, in the weighted adjacency graph $AG_\sigma(A_F^M, B_F^M)$.

Observe that the way the functions f and g have been defined, we have $|A_X| = |B_X| = |M|$, $c_{AG} = c_{AG_\sigma}$, $i_{AG} = i_{AG_\sigma}$, and thus

$$\begin{aligned} d_\sigma(A_F,B_F) &= 2|M| - w(M) - c_{AG_\sigma} - i_{AG_\sigma}/2 \\ &= 2|M| - |M| - c_{AG_\sigma} - i_{AG_\sigma}/2 \\ &= |M| - c_{AG_\sigma} - i_{AG_\sigma}/2 \\ &= |A_X| - c_{AG} - i_{AG}/2 \\ &= d(A_X,B_X)\,. \end{aligned}$$

Particularly, it is easy to see that we have

$$\begin{aligned} &\mathrm{opt}(\text{FFDCJ-DISTANCE}(A_F,B_F)) \\ &\quad = \mathrm{opt}((1,2)\text{-EXDCJ-DISTANCE}(A,B))\,. \end{aligned}$$

Therefore,

$$\begin{aligned} d(A_X,B_X) &= d_\sigma(A_F,B_F) \\ &\leq (1+\delta)\,\mathrm{opt}(\text{FFDCJ-DISTANCE}(A_F,B_F)) \\ &= (1+\delta)\,\mathrm{opt}((1,2)\text{-EXDCJ-DISTANCE}(A,B))\,, \end{aligned}$$

and Equation (3) holds by setting $\beta := 1$. □

Corollary 2. *There exists no polynomial-time algorithm for* FFDCJ-DISTANCE *with approximation factor better than* $1237/1236$*, unless* $P = NP$*.*

Proof. As shown in [8], $(1,2)$-EXDCJ-DISTANCE is NP-hard to approximate within a factor of $1237/1236 - \varepsilon$ for any $\varepsilon > 0$. Therefore, the result follows immediately from [8] and from the AP-reduction in the proof of Theorem 1. □

Since the weight plays an important role in d_σ, a matching with maximum weight, that is obviously maximal, could be a candidate for the design of an approximation algorithm for FFDCJ-DISTANCE. However, we can demonstrate that it is not possible to obtain such an approximation, with the following example.

Consider an integer $k \geq 1$ and let $A = \{(\circ\ 1\ -2\ \cdots\ (2k{-}1)\ -2k\ \circ)\}$ and $B = \{(\circ\ -(2k{+}1)\ (2k{+}2)\ \cdots\ -(2k{+}2k{-}1)\ (2k{+}2k)\ \circ)\}$ be two unichromosomal linear genomes. Observe that A and B have an even number of genes with alternating orientation. While A starts with a gene in direct orientation, B starts with a gene in reverse orientation. Now let σ be the normalized similarity measure between the genes of A and B, defined as follows:

$$\sigma(i,j) = \begin{cases} 1, & \text{for each } i \in \{1,2,\ldots,2k\} \text{ and } j = 2k+i\,; \\ 1-\varepsilon, & \text{for each } i \in \{1,3,\ldots,2k-1\} \text{ and } j = 2k+i+1, \text{ with } \varepsilon \in [0,1)\,; \\ 0, & \text{otherwise.} \end{cases}$$

Figure 5 shows $GS_\sigma(A,B)$ for $k = 3$ and σ as defined above.

There are several matchings in $GS_\sigma(A,B)$. We are interested in two particular maximal matchings:

- M^* is composed of all edges that have weight $1-\varepsilon$. It has weight $w(M^*) = (1-\varepsilon)|M^*| = (1-\varepsilon)\,k/2$. Its corresponding weighted adjacency graph $AG_\sigma(A^{M^*}, B^{M^*})$ has $|M^*| - 1$ cycles and two odd

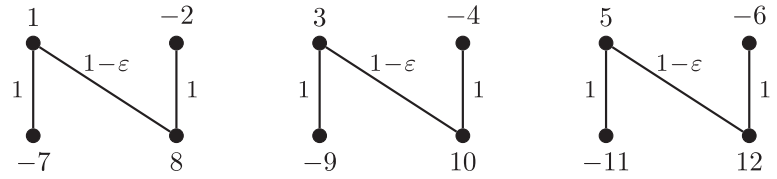

Figure 5 Gene similarity graph $GS_\sigma(A,B)$ for $k=3$.

paths, thus $\mathrm{d}_{\mathrm{DCJ}}(A^{M^*},B^{M^*})=0$. Consequently, we have $d_\sigma(A^{M^*},B^{M^*})=|M^*|-(1-\varepsilon)|M^*|=\varepsilon|M^*|$.

- M is composed of all edges that have weight 1. It is the only matching with the maximum weight $w(M)=|M|=k$. Its corresponding weighted adjacency graph $AG_\sigma(A^M,B^M)$ has two even paths, but no cycles or odd paths, giving $\mathrm{d}_{\mathrm{DCJ}}(A^M,B^M)=|M|$. Hence, $d_\sigma(A^M,B^M)=2|M|-|M|=|M|$.

Notice that $\mathrm{d}_{\mathrm{FFDCJ}}(A,B)\leq d_\sigma(A^{M^*},B^{M^*})$. Furthermore, since $|M|=2|M^*|$,

$$\frac{d_\sigma(A^M,B^M)}{d_\sigma(A^{M^*},B^{M^*})}=\frac{|M|}{\varepsilon|M^*|}=\frac{k}{\varepsilon\,k/2}=\frac{2}{\varepsilon}$$

and $2/\varepsilon\to+\infty$ when $\varepsilon\to 0$.

This shows that, for any genomes A and B, a matching of maximum weight in $GS_\sigma(A,B)$ can have d_σ arbitrarily far from the optimal solution and cannot give an approximation for FFDCJ-DISTANCE(A,B).

ILP to compute the family-free DCJ distance

We propose an integer linear program (ILP) formulation to compute the family-free DCJ distance between two given genomes. This formulation is a slightly different version of the ILP for the maximum cycle decomposition problem given by Shao *et al.* [10] to compute the DCJ distance between two given genomes with duplicate genes. Besides the cycle decomposition in a graph, as was made in [10], we also have to take into account maximal matchings in the gene similarity graph and their weights.

Let A and B be two genomes with extremity sets X_A and X_B, respectively, and let $G=GS_\sigma(A,B)$ be their gene similarity graph. The weight $w(e)$ of an edge e in G is also denoted by w_e. Let M be a maximal matching in G. For the ILP formulation, a weighted adjacency graph $H=AG_\sigma(A^M,B^M)$ is such that $V(H)=X_A\cup X_B$ and $E(H)$ has three types of edges: (*i*) *matching edges* that connect two extremities in different extremity sets, one in X_A and the other in X_B, if there exists one edge in M connecting these genes in G; the set of matching edges is denoted by E_m; (*ii*) *adjacency edges* that connect two extremities in the same extremity set if they are an adjacency; the set of adjacency edges is denoted by E_a; and (*iii*) *self edges* that connect two extremities of the same gene in an extremity set; the set of self edges is denoted by E_s. All edges in H are in $E_m\cup E_a\cup E_s=E(H)$. Matching edges have weights defined by the normalized similarity σ, all adjacency edges have weight 1, and all self edges have weight 0. Notice that any edge in G corresponds to two matching edges in H.

Now we describe the ILP. For each edge e in H, we create the binary variable x_e to indicate whether e will be in the final solution. We require first that each adjacency edge be chosen:

$$x_e=1\,,\qquad\forall\, e\in E_a\,.$$

We require then that, for each vertex in H, exactly one incident edge to it be chosen:

$$\sum_{uv\in E_m\cup E_s}x_{uv}=1\,,\ \forall\, u\in X_A\,,\ \text{and}\ \sum_{uv\in E_m\cup E_s}x_{uv}=1\,,\ \forall\, v\in X_B\,.$$

Then, we require that the final solution be consistent, meaning that if one extremity of a gene in A is assigned to an extremity of a gene in B, then the other extremities of these two genes have to be assigned as well:

$$x_{a^hb^h}=x_{a^tb^t}\,,\qquad\forall\, ab\in E(G)\,.$$

We also require that the matching be maximal. This can easily be ensured if we guarantee that at least one of the vertices connected by an edge in the gene similarity graph be chosen, which is equivalent to not allowing both of the corresponding self edges in the weighted adjacency graph be chosen:

$$x_{a^ha^t}+x_{b^hb^t}\leq 1\,,\qquad\forall\, ab\in E(G)\,.$$

To count the number of cycles, we use the same strategy as described in [10]. We first give an arbitrary index for each vertex in H such that $V(H)=\{v_1,v_2,\ldots,v_k\}$ with $k=|V(H)|$. For each vertex v_i, we define a variable y_i that labels v_i such that

$$0\leq y_i\leq i\,,\qquad 1\leq i\leq k\,.$$

We also require that adjacent vertices have the same label, forcing all vertices in the same cycle to have the same label:

$$y_i\leq y_j+i\cdot(1-x_e)\,,\quad\forall\, e=v_iv_j\in E(H)\,,$$
$$y_j\leq y_i+j\cdot(1-x_e)\,,\quad\forall\, e=v_iv_j\in E(H)\,.$$

We create a binary variable z_i, for each vertex v_i, to verify whether y_i is equal to its upper bound i:

$$i\cdot z_i\leq y_i\,,\qquad 1\leq i\leq k\,.$$

Since all the y_i variables in the same cycle have the same label but a different upper bound, only one of the y_i can be equal to its upper bound i. This means that for each cycle there can be only one z_i equal to 1, and the sum of all z_i variables is the total number of cycles in the adjacency graph.

In fact, it is possible to reduce the number of z_i variables. First, notice that each cycle always has vertices from both genomes. That means that if we label all vertices v_i starting with vertices of genome A first and then genome B, then the upper bounds for all y_is from genome A are smaller than the upper bounds for the y_is from genome B, and therefore no z_i from genome B will ever be 1, since in the same cycle there will be at least one y_i from genome A with a smaller upper bound. Then, all z_i corresponding to vertices of genome B may be discarded:

$$i \cdot z_i \leq y_i\,, \qquad 1 \leq i \leq |X_A|\,.$$

Finally, we set the objective function as follows:

$$\text{minimize} \quad 2\sum_{e\in E_m} x_e - \sum_{e\in E_m} w_e x_e - \sum_{1\leq i\leq |X_A|} z_i\,,$$

which is exactly the family-free DCJ distance $\mathrm{d}_{\mathrm{FFDCJ}}(A,B)$ as defined in Equations (1) and (2).

Simulations and experimental results

We performed some initial benchmarking experiments of the proposed ILP formulation. Therefore, we produced datasets using the Artificial Life Simulator (ALF) [17]. Genome sizes varied from 1000 to 3000 genes, where the gene lengths were generated according to a gamma distribution with shape parameter $k = 3$ and scale parameter $\theta = 133$. A birth-death tree with 10 leaves was generated, with PAM distance of 100 from the root to the deepest leaf. For the amino acid evolution, the WAG substitution model with default parameters was used, with Zipfian indels at a rate of 0.000005. For structural evolution, gene duplications and gene losses were applied with a rate of 0.001 and reversals and translocations with a rate of 0.0025. To test different proportions of rearrangement events, we also simulated datasets where the structural evolution ratios had a 2- and 5-fold increase.

To solve the ILPs, we ran the CPLEX Optimizer on the 45 pairwise comparisons of each simulated dataset. All simulations were run in parallel on a cluster consisting of machines with an Intel(R) Xeon(R) E7540 CPU, with 48 cores and as many as 2 TB of memory, but for each individual CPLEX run only 4 cores and 2 GB of memory were allocated. The results are summarized in Table 1.

The family-free DCJ similarity

For a given matching M in $GS_\sigma(A,B)$, a formula for the similarity s_σ of the reduced genomes A^M and B^M was first proposed in [12] only considering the cycles of $AG_\sigma(A^M,B^M)$. Here we extend this formula to consider all components of the weighted adjacency graph. Again, let $\mathcal{C}$, $\mathcal{I}$, and $\mathcal{P}$ represent the sets of components in $AG_\sigma(A^M,B^M)$ that are cycles, odd paths and even paths, respectively. Furthermore, $w(C) = \sum_{e\in C}\sigma(e)$ is the sum of the weights of all the edges in a component C. Then the similarity s_σ is the normalized total weight of all components:

$$s_\sigma(A^M,B^M) = \sum_{C\in\mathcal{C}}\left(\frac{w(C)}{|C|}\right) + \sum_{C\in\mathcal{I}}\left(\frac{w(C)}{|C|+1}\right) + \sum_{C\in\mathcal{P}}\left(\frac{w(C)}{|C|+2}\right).$$

Here our goal is to study the problem of computing the family-free DCJ similarity, i.e., to find a matching in $GS_\sigma(A,B)$ that maximizes s_σ. Similarly to the distance, the behaviour of the similarity does not correlate with the size of the matching. In other words, smaller matchings, that possibly discard gene assignments, can lead to higher similarities.

An approach for solving this problem was proposed in [12], following the one in [11] for gene adjacencies. It consists of a parameterized similarity function $\mathcal{F}_\alpha$ in which the user-controlled parameter α is a real number between 0 and 1:

$$\mathcal{F}_\alpha(A^M,B^M) = \alpha\cdot s_\sigma(A^M,B^M) + (1-\alpha)\cdot w(M)\,,$$

where, as above, $w(M) = \sum_{e\in M} w(e)$ is the sum of the edge weights of the matching M.

Observe that the parameter α can be adjusted in favor of gene similarity when α is closer to 0, or in favor of genome organization similarity, when α is closer to 1. The closer

Table 1 ILP running-time results for datasets with different genome sizes and evolutionary rates

	1000 genes			2000 genes			3000 genes		
	$r=1$	$r=2$	$r=5$	$r=1$	$r=2$	$r=5$	$r=1$	$r=2$	$r=5$
Finished	35/45	10/45	2/45	45/45	9/45	1/45	45/45	7/45	3/45
Avg. Time (s)	99.66	6.97	0.53	0.47	0.70	3.31	0.45	2.03	213.15
Avg. Gap (%)	0.3	3.0	4.3	0	3.6	6.5	0	5.3	4.8

Each dataset has 10 genomes, totalling 45 pairwise comparisons. Maximum running time was set to 60 minutes. For each dataset, the number of runs is shown that found an optimal solution within the allowed time and their average running time in seconds. For the runs that did not finish, the last row shows the relative gap between the upper bound and the current solution. Rate $r = 1$ means the default rate for ALF evolution, and $r = 2$ and $r = 5$ mean 2-fold and 5-fold increase for the gene duplication, gene deletion and rearrangement rates.

the parameter α is to 0, the closer we are to the problem of finding a maximum weighted matching in the gene similarity graph $GS_\sigma(A,B)$. On the other hand, the closer α is to 1, the closer we are to the problem of computing $s_\sigma(A^M,B^M)$. A drawback of this model is that the weights of edges actually appear in both terms of the equation. Furthermore, it remains the problem of finding the "best" value for α.

Here, instead of adopting the parameter α, we restrict the similarity to *maximal matchings* only, ensuring that no pair of genes with positive similarity score is simply discarded, even though it might decrease the overall similarity. We then have the following optimization problem:

Problem FFDCJ-SIMILARITY(A,B): Given genomes A and B and their gene similarities σ, calculate their family-free DCJ similarity

$$s_{\text{FFDCJ}}(A,B) = \max_{M\in\mathbb{M}}\{s_\sigma(A^M,B^M)\}\,,$$

where $\mathbb{M}$ is the set of all maximal matchings in $GS_\sigma(A,B)$.

Complexity of the family-free DCJ similarity

We have the following result to the family-free DCJ similarity.

Theorem 3. *Problem* FFDCJ-SIMILARITY *is NP-hard, even if the maximum degrees in the two partitions of the gene similarity graph are respectively one and two.*

Proof. We use the Cook reduction, which is a polynomial time transformation, from $(1,2)$-EXDCJ-DISTANCE to FFDCJ-SIMILARITY.

Let A and B be any instance of $(1,2)$-EXDCJ-DISTANCE and let k be a positive integer, with $k \leq |A|$, where $|A|$ is the number of genes of a genome A. We suppose, without loss of generality, that A and B are circular multichromosomal genomes. We must construct a pair of circular genomes A_F and B_F, a normalized similarity measure σ for genes in A_F and B_F, and a positive integer $k' \leq |A_\text{F}|$ such that the family-free DCJ similarity of A_F and B_F is at least k' if and only if the exemplar DCJ distance of genomes A and B is at most k.

The construction of $A_\text{F}, B_\text{F}, \sigma$, and k' is similar to the transformation f in (AP1) of the proof of Theorem 1. Let $\mathcal{G}$ be the underlying gene set, such that each gene in $\mathcal{G}$ occurs at most once in A and at most twice in B. Let the genes of A be denoted $a_1, a_2, \ldots, a_{|A|}$ and the genes of B be denoted $b_1, b_2, \ldots, b_{|B|}$. Then A_F and B_F are copies of A and B, respectively, except that symbol a_i in A_F is relabeled by i, keeping its orientation, and b_j in B_F is relabeled by $j + |A|$, also keeping its orientation. The normalized similarity measure σ for genes in A_F and B_F is defined as $\sigma(i,k) = 1$ for i in A_F and k in B_F, such that a_i is in A, b_j is in B, a_i and b_j are in the same gene family, and $k = j + |A|$. Otherwise, $\sigma(i,k) = 0$. It is easy to see that this construction can be accomplished in poynomial time.

Now we must show that the family-free DCJ similarity of A_F and B_F is at least k' if and only if the exemplar DCJ distance of genomes A and B is at most k. Let $n = |A|$.

Suppose first that M is a matching in the gene similarity graph $GS_\sigma(A_\text{F}, B_\text{F})$ such that $s_\sigma(A_\text{F}^M, B_\text{F}^M) \geq k'$. For each edge (i,k) in M, we add symbols a_i to A_X and b_j to B_X, where $j = k - |A|$. Notice that $|M| = |A_\text{F}^M| = |A_\text{X}| = |A|$. Then, since the genomes in both instances are circular and the edge weights in the gene similarity graph of A_F and B_F are all one, we have

$$k' \leq s_\sigma(A_\text{F}^M, B_\text{F}^M) = \sum_{C\in\mathcal{C}} \frac{w(C)}{|C|} = c_{AG_\sigma} = c_{AG} = |M| - d(A_\text{X}, B_\text{X})\,,$$

where c_{AG} is the number of cycles in the adjacency graph $AG(A_\text{X}, B_\text{X})$. Thus, by setting $k = n - k'$, we have

$$d(A_\text{X}, B_\text{X}) \leq n - k' = k\,.$$

On the other hand, suppose that for an instance (A,B) of $(1,2)$-EXDCJ-DISTANCE we have exemplar genomes A_X and B_X such that $d(A_\text{X}, B_\text{X}) \leq k$. The exemplar genomes A_X and B_X induce a matching M in the gene similarity graph $GS_\sigma(A_\text{F}, B_\text{F})$ and, once again, since the genomes in both instances are circular and the edge weights in $GS_\sigma(A_\text{F}, B_\text{F})$ are all one, we have

$$\begin{aligned} k \geq d(A_\text{X}, B_\text{X}) &= n - c_{AG} = n - c_{AG_\sigma} = n - \sum_{C\in\mathcal{C}} \frac{w(C)}{|C|} \\ &= n - s_\sigma\left(A_\text{F}^M, B_\text{F}^M\right)\,, \end{aligned}$$

where c_{AG} is the number of cycles in the adjacency graph $AG(A_\text{X}, B_\text{X})$. By setting $k' = n - k$ we have

$$s_\sigma(A_\text{F}^M, B_\text{F}^M) \geq n - k = k'\,.$$

□

Conclusion

In this paper, we have defined a new distance measure for two genomes that is motivated by the double cut and join model, while not relying on gene annotations in form of gene families. In case gene families are known and each family has exactly one member in each of the two genomes, this distance equals the family-based DCJ distance and thus can be computed in linear time. In the general case, however, it is NP-hard and even hard to approximate. Nevertheless, we could give an integer linear program for the exact computation of the distance that is fast enough to be applied to realistic problem instances. Similar theoretical results hold for the family-free DCJ similarity measure, which is NP-hard.

The family-free model has many potentials when gene family assignments are not available or ambiguous, in fact it can even be used to improve family assignments [18]. The work presented in this paper is another step in this direction.

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

Authors' contributions
FVM, PF, MDVB and JS developed the theoretical FFDCJ model. FVM worked out the details of the complexity results. PF developed the ILP and ran the experimental results. All authors read and approved the final manuscript.

Acknowledgements
We would like to thank Tomáš Vinař who suggested that the NP-hardness of FFDCJ-DISTANCE could be proven via a reduction from the exemplar distance problem. FVM and MDVB are funded from the Brazilian research agency CNPq grants Ciência sem Fronteiras Postdoctoral Scholarship 245267/2012-3 and PROMETRO 563087/2010-2, respectively.

Author details
[1]Faculdade de Computação, Universidade Federal de Mato Grosso do Sul, Avenida Costa e Silva, s-n, 79070-900 Campo Grande, MS, Brazil. [2]Technische Fakultät and CeBiTec, Universität Bielefeld, Universitätsstr. 25, 33615 Bielefeld, Germany. [3]Inmetro – Instituto Nacional de Metrologia, Qualidade e Tecnologia, Av. Nossa Senhora das Graças, 50, 25250-020 Duque de Caxias, RJ, Brazil.

Received: 10 February 2015 Accepted: 13 March 2015
Published online: 01 April 2015

References

1. Sankoff D. Edit distance for genome comparison based on non-local operations. In: Proc. of CPM 1992. LNCS, vol. 644. Heidelberg: Springer Verlag; 1992. p. 121–35.
2. Bergeron A, Mixtacki J, Stoye J. A unifying view of genome rearrangements. In: Proc. of WABI 2006. LNBI, vol. 4175. Heidelberg: Springer Verlag; 2006. p. 163–73.
3. Bafna V, Pevzner P. Genome rearrangements and sorting by reversals. In: Proc. of FOCS 1993; 1993. p. 148–57.
4. Hannenhalli S, Pevzner P. Transforming men into mice (polynomial algorithm for genomic distance problem). In: Proc. of FOCS 1995; 1995. p. 581–92.
5. Yancopoulos S, Attie O, Friedberg R. Efficient sorting of genomic permutations by translocation, inversion and block interchanges. Bioinformatics. 2005;21(16):3340–6.
6. Sankoff D. Genome rearrangement with gene families. Bioinformatics. 1999;15(11):909–17.
7. Bryant D. The complexity of calculating exemplar distances In: Sankoff D, Nadeau JH, editors. Comparative Genomics. Dortrecht: Kluwer Academic Publishers; 2000. p. 207–11.
8. Bulteau L, Jiang M. Inapproximability of (1,2)-exemplar distance. IEEE/ACM Trans Comput Biol Bioinf. 2013;10(6):1384–90.
9. Angibaud S, Fertin G, Rusu I, Thévenin A, Vialette S. On the approximability of comparing genomes with duplicates. J Graph Algorithms Appl. 2009;13(1):19–53.
10. Shao M, Lin Y, Moret B. An exact algorithm to compute the DCJ distance for genomes with duplicate genes. In: Proc. of RECOMB 2014. LNBI, vol. 8394. Heidelberg: Springer Verlag; 2014. p. 280–92.
11. Dörr D, Thévenin A, Stoye J. Gene family assignment-free comparative genomics. BMC Bioinformatics. 2012;13(Suppl 19):3.
12. Braga MDV, Chauve C, Dörr D, Jahn K, Stoye J, Thévenin A, et al. The potential of family-free genome comparison In: Chauve C, El-Mabrouk N, Tannier E, editors. Models and Algorithms for Genome Evolution, Chap. 13. London: Springer; 2013. p. 287–307.
13. Martinez FV, Feijão P, Braga MDV, Stoye J. On the family-free DCJ distance. In: Proc. of WABI 2014. LNBI, vol. 8701. Heidelberg: Springer Verlag; 2014. p. 174–86.
14. Braga MDV, Stoye J. The solution space of sorting by DCJ. J Comp Biol. 2010;17(9):1145–65.
15. Feijão P, Meidanis J, SCJ: A breakpoint-like distance that simplifies several rearrangement problems. IEEE/ACM Trans Comput Biol Bioinf. 2011;8(5): 1318–29.
16. Ausiello G, Protasi M, Marchetti-Spaccamela A, Gambosi G, Crescenzi P, Kann V. Complexity and approximation: combinatorial optimization problems and their approximability properties. Heidelberg: Springer; 1999.
17. Dalquen DA, Anisimova M, Gonnet GH, Dessimoz C. ALF–a simulation framework for genome evolution. Mol Biol Evol. 2012;29(4):1115–23.
18. Lechner M, Hernandez-Rosales M, Doerr D, Wieseke N, Thévenin A, Stoye J, et al. Orthology detection combining clustering and synteny for very large datasets. PLOS ONE. 2014;9(8):107014.