# On the Family-Free DCJ Distance

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

[1] Faculdade de Computação, Universidade Federal de Mato Grosso do Sul, Brazil
[2] Technische Fakultät and CeBiTec, Universität Bielefeld, Germany
[3] Inmetro – Instituto Nacional de Metrologia, Qualidade e Tecnologia, Brazil

**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 were already proposed 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 similarity between genes. We propose a new family-free DCJ distance, prove that the family-free DCJ distance problem is APX-hard, and provide an integer linear program to its solution.

## 1 Introduction

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 [14].

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 [3,4,12,17]. However, when gene duplications are allowed, the problem is more intrincate and all approaches proposed so far are NP-hard, see for instance [1,7,8,15,16].

D. Brown and B. Morgenstern (Eds.): WABI 2014, LNBI 8701, pp. 174–186, 2014.

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 [5, 10]. 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 [17]. 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 in Section 4 a hardness result, before giving a linear programming solution and showing its feasibility for practical problem instances in Section 5. Section 6 concludes.

## 2 Preliminaries

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 $B$.

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 $2^h$.

### 2.1 Adjacency Graph and Family-Based DCJ Distance

In the family-based setting we are given two genomes $A$ and $B$ with 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.

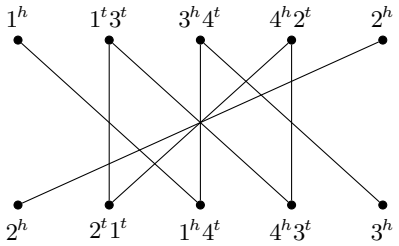


**Fig. 1.** The adjacency graph for the two unichromosomal and linear genomes $A = \{(\circ\ {-1}\ 3\ 4\ 2\ \circ)\}$ and $B = \{(\circ\ {-2}\ 1\ 4\ 3\ \circ)\}$

The family-based DCJ distance $\mathrm{d}_{\mathrm{DCJ}}$ 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 $AG(A, B)$ [4]:

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

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 $AG(A, B)$.

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 $\mathrm{d}_{\mathrm{DCJ}}(A, B) = 4 - 1 - 2/2 = 2$.

The formula for $\mathrm{d}_{\mathrm{DCJ}}(A, B)$ 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 [6,11] 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:

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

### 2.2 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 $\sigma$ in the so called *gene similarity graph* [5], 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.

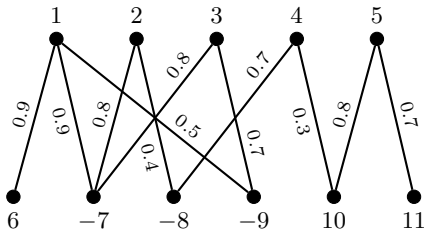


**Fig. 2.** A possible gene similarity graph for the two unichromosomal linear genomes $A = \{(\circ\ 1\ 2\ 3\ 4\ 5\ \circ)\}$ and $B = \{(\circ\ 6\ {-7}\ {-8}\ {-9}\ 10\ 11\ \circ)\}$

## 3 Reduced Genomes and Family-Free DCJ Distance

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 $s(a, M) = s(b, M) = i$. 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 $s(a, M)$, 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 $s(b, M)$, preserving its orientation. Observe that the set of genes in $A^M$ and in $B^M$ is $\mathcal{G}(M) = \{s(g, M) : g \text{ is saturated by the matching } M\} = \{1, 2, \ldots, n\}$.

### 3.1 The Weighted Adjacency Graph of Reduced Genomes

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.

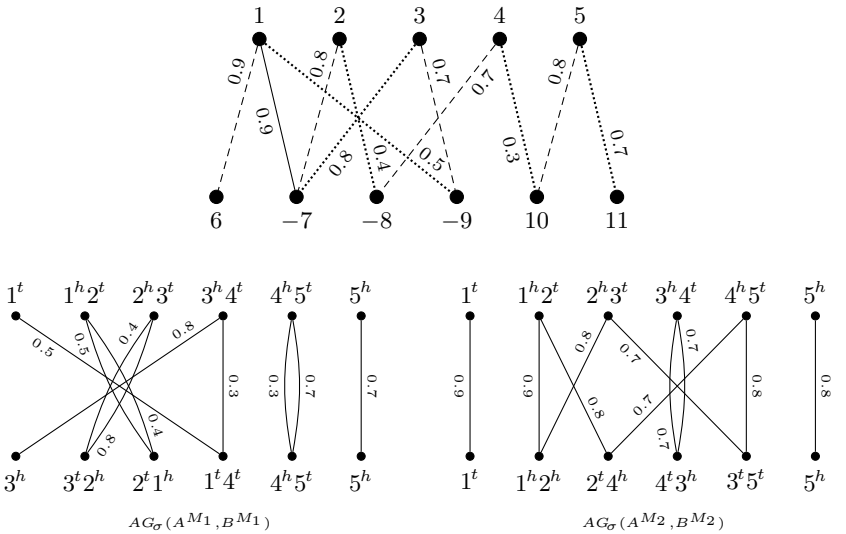


**Fig. 3.** Considering the same genomes $A = \{(\circ\ 1\ 2\ 3\ 4\ 5\ \circ)\}$ and $B = \{(\circ\ 6\ -7\ -8\ -9\ 10\ 11\ \circ)\}$ as in Figure 2, let $M_1$ (dotted edges) and $M_2$ (dashed edges) be two distinct matchings in $GS_\sigma(A, B)$, shown in the upper part. The two resulting weighted adjacency 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.

### 3.2 The Weighted DCJ Distance of Reduced Genomes

Based on the weighted adjacency graph, in [5] a family-free DCJ similarity measure has been proposed. To be more consistent with the comparative genomics literature, where distance measures are more common than similarities, here we propose a general family-free DCJ distance. Moreover, edge weights are treated in a way that, when all weights are equal to 1, the definition falls back to the (unweighted) family-based DCJ distance.

To define the distance measure, we consider the components of the graph $AG_\sigma(A^M, B^M)$ separately, similarly to the approach described in Section 2.1 for the family-based model. Now, 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 components, the resulting distance for a certain matching $M$ is computed as follows:

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

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

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.

### 3.3 The Family-Free DCJ Distance

Our goal in the remainder of this paper 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 $d_\sigma$. 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. 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 $\sigma$, calculate their family-free DCJ distance
>
> $$\mathrm{d}_{\mathrm{FFDCJ}}(A, B) = \min_{M \in \mathbb{M}} \{d_\sigma(A^M, B^M)\} ,$$
>
> where $\mathbb{M}$ is the set of all maximal matchings in $GS_\sigma(A, B)$.

## 4 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* [8, 15]:

> **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 $\mathrm{d}_{\mathrm{DCJ}}(A', B')$ 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.*

*Proof.* Using notation from [2] (Chapter 8), we give an AP-reduction $(f, g, \beta)$ from $(1, 2)$-EXDCJ-DISTANCE to FFDCJ-DISTANCE as follows:

Algorithm $f$ receives as input a positive rational number $\delta$ 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 $(A', B') = f(\delta, (A, B))$ 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 $A'$ and $B'$ are copies of $A$ and $B$, respectively, except that symbol $a_i$ in $A'$ is relabeled by $i$, keeping its orientation, and $b_j$ in $B'$ is relabeled by $j + |A|$, also keeping its orientation. Furthermore, the similarity $\sigma$ for genes in $A'$ and $B'$ is defined as $\sigma(i, k) = 1$ for $i$ in $A'$ and $k$ in $B'$, 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, $\sigma(i, k) = 0$. Figure 4 gives an example of a $GS_\sigma(A', B')$ for this construction.

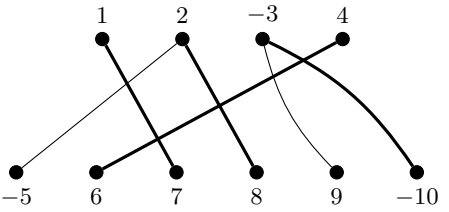


**Fig. 4.** Gene similarity graph $GS_\sigma(A', B')$ constructed from the input genomes $A = \{(\circ\ a\ c\ {-b}\ d\ \circ)\}$ and $B = \{(\circ\ {-c}\ \ d\ a\ c\ b\ {-b}\ \circ)\}$ of $(1,2)$-EXDCJ-DISTANCE, where all edge weights are 1. Highlighted edges represent a maximal matching in $GS_\sigma(A',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

Algorithm $g$ receives as input a positive rational number $\delta$, an instance $(A, B)$ of $(1,2)$-EXDCJ-DISTANCE and a solution $M'$ of FFDCJ-DISTANCE, and transforms $M'$ into a solution $(A_{\text{X}}, B_{\text{X}})$ of $(1,2)$-EXDCJ-DISTANCE. This is a simple construction: for each edge $(i, k)$ in $M'$ we add symbols $a_i$ to $A_{\text{X}}$ and $b_j$ to $B_{\text{X}}$, where $j = k - |A|$. For the example of Figure 4, a matching $M' = \{(1,7), (2,8), (-3,-10), (4,6)\}$, which is a solution to FFDCJ-DISTANCE$(A', B')$, is transformed by $g$ into the genomes $A_{\text{X}} = \{(\circ\ a_1\ a_2\ a_3\ a_4\ \circ)\} = \{(\circ\ a\ c\ {-b}\ d\ \circ)\}$ and $B_{\text{X}} = \{(\circ\ b_2\ b_3\ b_4\ b_6\ \circ)\} = \{(\circ\ d\ a\ c\ {-b}\ \circ)\}$, which is a solution to $(1,2)$-EXDCJ-DISTANCE$(A, B)$.

Notice that for any positive rational number $\delta$, functions $f$ and $g$ are polynomial time algorithms on the size of their respective instances. Let $A_{\text{X}} := A$ and let $B_{\text{X}}$ be an exemplar genome of $B$, such that $(A_{\text{X}}, B_{\text{X}}) = g(\delta, (A, B), M')$. Denote by $c_{AG}$ and $i_{AG}$ the number of cycles and odd paths in $AG(A_{\text{X}}, B_{\text{X}})$, and by $c_{AG_\sigma}$ and $i_{AG_\sigma}$ the number of cycles and odd paths in $AG_\sigma(A^{M'}, B^{M'})$. Observe that we have $|A_{\text{X}}| = |B_{\text{X}}| = |M'|$, $c_{AG} = c_{AG_\sigma}$, $i_{AG} = i_{AG_\sigma}$, and thus

$$\begin{aligned} d_\sigma(A', B') &= 2|M'| - w(M') - c_{AG_\sigma} - i_{AG_\sigma}/2 \\ &= |A_{\text{X}}| - c_{AG} - i_{AG}/2 \\ &= d(A_{\text{X}}, B_{\text{X}})\,, \end{aligned}$$

that is, $\text{opt}(\text{FFDCJ-DISTANCE}(A', B')) = \text{opt}((1,2)\text{-EXDCJ-DISTANCE}(A, B))$.

Therefore, $d_\sigma(A', B') \leq (1+\delta)\,\text{opt}((1,2)\text{-EXDCJ-DISTANCE}(A, B))$ for any positive $\delta$, and the last condition for the AP-reduction holds by setting $\beta := 1$:

$$d(A_{\text{X}}, B_{\text{X}}) \leq (1 + \beta\delta)\,\text{opt}((1,2)\text{-EXDCJ-DISTANCE}(A, B))\,.$$

□

**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 $d_\sigma$, 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 $\sigma$ 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 $\sigma$ as defined above.

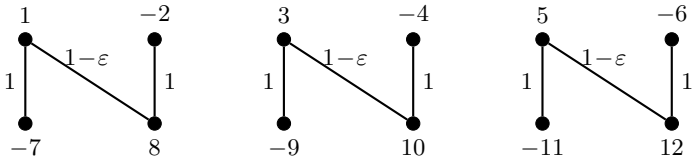


**Fig. 5.** Gene similarity graph $GS_\sigma(A,B)$ for $k = 3$

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 $w(M^*) = (1{-}\varepsilon)|M^*|$. Its corresponding weighted adjacency graph $AG_\sigma(A^{M^*}, B^{M^*})$ has $|M^*|-1$ cycles and two odd 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 $w(M) = |M|$. 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 $|M| = 2|M^*|$,

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

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 $d_\sigma$ arbitrarily far from the optimal solution and cannot give an approximation for FFDCJ-DISTANCE$(A, B)$.

## 5 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.* [16] to compute the DCJ distance between two given genomes with duplicate genes. Besides the cycle decomposition in a graph, as was made in [16], 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 $\sigma$, 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:

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

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:

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

We also require that the matching be maximal. It can be easily ensured if we garantee 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:

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

To count the number of cycles, we use the same strategy as described in [16]. 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 all vertices in the same cycle in the solution have the same label:

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

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

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

Notice that the way as variables $z_i$ were defined, they count the number of cycles in $H$ [16].

Finally, we set the objective function as follows:

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

which is exactly the family-free DCJ distance $\mathrm{d}_{\mathrm{FFDCJ}}(A, B)$ as defined in Section 3.

We performed some initial simulated experiments of our integer linear program formulation. We produced some datasets using the Artificial Life Simulator (ALF) [9]. 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 0.001 rate, with a 0.0025 rate for reversals and translocations. To test different ration 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[1] 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 on Table 1.

**Table 1.** ILP results for datasets with different genome sizes and evolutionary rates. Each dataset has 10 genomes, totalling 45 pairwise comparisons. Maximum running time was set to 20 minutes. For each dataset, it is shown the number of runs that found an optimal solution in time and their average running time. For the runs that did not finish, the last row shows the 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.

| | 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 | 45/45 | 22/45 | 6/45 | 45/45 | 9/45 | 1/45 | 45/45 | 7/45 | 3/45 |
| Avg. Time (s) | 0.66 | 11.09 | 24.26 | 1.29 | 2.76 | 16.97 | 2.24 | 16.36 | 36.01 |
| Avg. Gap (%) | 0 | 1.08 | 3.9 | 0 | 1.93 | 12.4 | 0 | 3.9 | 6.03 |

## 6 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, the 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.

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 [13]. The work presented in this paper is another step in this direction.

**Acknowledgments.** 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.

[1] `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`

## References


1. Angibaud, S., Fertin, G., Rusu, I., Thévenin, A., Vialette, S.: On the approximability of comparing genomes with duplicates. J. Graph Algorithms Appl. 13(1), 19–53 (2009)
2. Ausiello, G., Protasi, M., Marchetti-Spaccamela, A., Gambosi, G., Crescenzi, P., Kann, V.: Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties. Springer (1999)
3. Bafna, V., Pevzner, P.: Genome rearrangements and sorting by reversals. In: Proc. of FOCS 1993, pp. 148–157 (1993)
4. Bergeron, A., Mixtacki, J., Stoye, J.: A unifying view of genome rearrangements. In: Bücher, P., Moret, B.M.E. (eds.) WABI 2006. LNCS (LNBI), vol. 4175, pp. 163–173. Springer, Heidelberg (2006)
5. Braga, M.D.V., Chauve, C., Dörr, D., Jahn, K., Stoye, J., Thévenin, A., Wittler, R.: The potential of family-free genome comparison. In: Chauve, C., El-Mabrouk, N., Tannier, E. (eds.) Models and Algorithms for Genome Evolution, ch. 13, pp. 287–307. Springer (2013)
6. Braga, M.D.V., Stoye, J.: The solution space of sorting by DCJ. J. Comp. Biol. 17(9), 1145–1165 (2010)
7. Bryant, D.: The complexity of calculating exemplar distances. In: Sankoff, D., Nadeau, J.H. (eds.) Comparative Genomics, pp. 207–211. Springer, Netherlands (2000)
8. Bulteau, L., Jiang, M.: Inapproximability of (1,2)-exemplar distance. IEEE/ACM Trans. Comput. Biol. Bioinf. 10(6), 1384–1390 (2013)
9. Dalquen, D.A., Anisimova, M., Gonnet, G.H., Dessimoz, C.: ALF–a simulation framework for genome evolution. Mol. Biol. Evol. 29(4), 1115–1123 (2012)
10. Dörr, D., Thévenin, A., Stoye, J.: Gene family assignment-free comparative genomics. BMC Bioinformatics 13(Suppl 19), S3 (2012)
11. Feijão, P., Meidanis, J.: SCJ: A breakpoint-like distance that simplifies several rearrangement problems. IEEE/ACM Trans. Comput. Biol. Bioinf. 8(5), 1318–1329 (2011)
12. Hannenhalli, S., Pevzner, P.: Transforming men into mice (polynomial algorithm for genomic distance problem). In: Proc. of FOCS 1995, pp. 581–592 (1995)
13. Lechner, M., Hernandez-Rosales, M., Doerr, D., Wieseke, N., Thévenin, A., Stoye, J., Hartmann, R.K., Prohaska, S.J., Stadler, P.F.: Orthology detection combining clustering and synteny for very large datasets (unpublished manuscript)
14. Sankoff, D.: Edit distance for genome comparison based on non-local operations. In: Apostolico, A., Galil, Z., Manber, U., Crochemore, M. (eds.) CPM 1992. LNCS, vol. 644, pp. 121–135. Springer, Heidelberg (1992)
15. Sankoff, D.: Genome rearrangement with gene families. Bioinformatics 15(11), 909–917 (1999)
16. Shao, M., Lin, Y., Moret, B.: An exact algorithm to compute the DCJ distance for genomes with duplicate genes. In: Sharan, R. (ed.) RECOMB 2014. LNCS, vol. 8394, pp. 280–292. Springer, Heidelberg (2014)
17. Yancopoulos, S., Attie, O., Friedberg, R.: Efficient sorting of genomic permutations by translocation, inversion and block interchanges. Bioinformatics 21(16), 3340–3346 (2005)