

Natural Family-Free Genomic Distance

Diego P. Rubert 

Faculdade de Computação, Universidade Federal de Mato Grosso do Sul, Campo Grande, Brazil
diego@facom.ufms.br

Fábio V. Martinez 

Faculdade de Computação, Universidade Federal de Mato Grosso do Sul, Campo Grande, Brazil
fhvm@facom.ufms.br

Marília D. V. Braga¹ 

Faculty of Technology and Center for Biotechnology (CeBiTec), Bielefeld University, Germany
mbraga@cebitec.uni-bielefeld.de

Abstract

A classical problem in comparative genomics is to compute the rearrangement distance, that is the minimum number of large-scale rearrangements required to transform a given genome into another given genome. While the most traditional approaches in this area are *family-based*, i.e., require the classification of DNA fragments of both genomes into *families*, more recently an alternative model was proposed, which, instead of family classification, simply uses the *pairwise similarities* between DNA fragments of both genomes to compute their rearrangement distance. This model represents structural rearrangements by the generic *double cut and join* (DCJ) operation and is then called *family-free DCJ distance*. It computes the DCJ distance between the two genomes by searching for a matching of their genes based on the given pairwise similarities, therefore helping to find gene homologies. The drawback is that its computation is NP-hard. Another point is that the family-free DCJ distance must correspond to a maximal matching of the genes, due to the fact that unmatched genes are just ignored: maximizing the matching prevents the *free lunch* artifact of having empty or almost empty matchings giving the smaller distances.

In this paper, besides DCJ operations, we allow content-modifying operations of *insertions* and *deletions* of DNA segments and propose a new and more general family-free genomic distance. In our model we use the pairwise similarities to assign weights to both matched and unmatched genes, so that an optimal solution does not necessarily maximize the matching. Our model then results in a *natural family-free genomic distance*, that takes into consideration all given genes and has a search space composed of matchings of any size. We provide an efficient ILP formulation to solve it, by extending the previous formulations for computing family-based genomic distances from Shao et al. (*J. Comput. Biol.*, 2015) and Bohnenkämper et al. (*Proc. of RECOMB*, 2020). Our experiments show that the ILP can handle not only bacterial genomes, but also fungi and insects, or sets of chromosomes of mammals and plants. In a comparison study of six fruit fly genomes, we obtained accurate results.

2012 ACM Subject Classification Theory of computation → Linear programming

Keywords and phrases Comparative genomics, Genome rearrangement, DCJ-indel distance

Digital Object Identifier 10.4230/LIPIcs.WABI.2020.3

Related Version <http://arxiv.org/abs/2007.03556>

Supplementary Material The ILP for computing the natural family-free DCJ-indel distance can be downloaded from our GitLab server at <https://gitlab.ub.uni-bielefeld.de/gi/gen-diff>.

Acknowledgements We thank the anonymous reviewers for their valuable comments.

¹ Corresponding author



1 Introduction

Genomes are subject to mutations or *rearrangements* in the course of evolution. A classical problem in comparative genomics is to compute the rearrangement *distance*, that is the minimum number of large-scale rearrangements required to transform a given genome into another given genome [21]. Typical large-scale rearrangements change the number of chromosomes, and/or the positions and orientations of DNA segments. Examples of such *structural* rearrangements are inversions, translocations, fusions and fissions. One might also need to consider rearrangements that modify the content of a genome, such as insertions and deletions (collectively called *indels*) of DNA segments.

In order to study the rearrangement distance, 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 fragments in both genomes into *families*, so that two fragments in the same family are said to be equivalent. This setting is said to be *family-based*. Without duplications, that is, with the additional restriction that each family occurs at most once in each genome, many polynomial models have been proposed to compute the genomic distance [3, 6, 13, 24, 25]. However, when duplications are allowed the problem is more intricate and all approaches proposed so far are NP-hard, see for instance [2, 7, 8, 18, 22, 23].

The required pre-classification of DNA fragments into families is a drawback of the family-based approaches. Moreover, even with a careful pre-processing, it is not always possible to classify each fragment unambiguously into a single family. Due to these facts, an alternative to the family-based setting was proposed and consists in studying the rearrangement distance without prior family assignment. Instead of families, the *pairwise similarities* between fragments is directly used [5, 12]. By letting structural rearrangements be represented by the generic *double cut and join* (DCJ) operation [24], a first family-free genomic distance, called family-free DCJ distance, was already proposed [16]. Its computation helps to match occurrences of duplicated genes and find homologies, but unmatched genes are simply ignored.

In the family-based setting, the mentioned approaches that handle duplications either require the compared genomes to be *balanced* (that is, have the same number of occurrences of each family) [18, 23] or adopt some approach to match genes, ignoring unmatched genes [8, 22]. Recently, a new family-based approach was proposed, allowing each family to occur any number of times in each genome and integrating DCJ operations and indels in a *DCJ-indel* distance formula [4]. For its computation, that is NP-hard, an efficient ILP was proposed.

Here we adapt the approach mentioned above and give an ILP formulation to compute a new family-free DCJ-indel distance. In the family-based approach from [4] as well as in the family-free DCJ distance proposed in [16], the search space needs to be restricted to candidates that maximize the number of matched genes, in order to avoid the *free lunch* artifact of having empty or almost empty matchings giving the smaller distances [25]. In our formulation we use the pairwise similarities to assign weights to matched and unmatched genes, so that, for the first time, an optimal solution does not necessarily maximize the number of matched genes. For the fact that our model takes into consideration all given genes and has a search space composed of matchings of any size, we call it *natural family-free genomic distance*. Our simulated experiments show that our ILP can handle not only bacterial genomes, but also complete genomes of fungi and insects, or sets of chromosomes of mammals and plants. We use our implementation to generate pairwise distances and reconstruct the phylogeny of six species of fruit flies from the genus *Drosophila*, obtaining accurate results.

This paper is organized as follows. In Section 2 we give some basic definitions and previous results that are essential for the approach presented here. In Section 3 we define the new *natural* family-free DCJ-indel distance. In Section 4 we describe the optimization approach for computing the family-free DCJ-indel distance with the help of the *family-free relational diagram*. In Section 5 we present the ILP formulation and the experimental results. Finally, Section 6 concludes the text.

2 Preliminaries

We call *marker* an oriented DNA fragment. A *chromosome* is composed of markers and can be linear or circular. A marker m in a chromosome can be represented by the symbol m itself, if it is read in direct orientation, or the symbol \overline{m} , if it is read in reverse orientation. We concatenate all markers of a chromosome Z in a string z , which can be read in any of the two directions. If Z is circular, we can start to read it at any marker and the string z is flanked by parentheses. A set of chromosomes comprises a *genome*. As an example, let $A = \{\overline{61784}, 3\overline{52}\}$ be a genome composed of two linear chromosomes. A genome can be transformed or *sorted* into another genome with the following types of mutations.

1. **DCJ operations modify the organization of a genome:** A *cut* performed on a genome A separates two adjacent markers of A . A *double-cut and join* or *DCJ* applied on a genome A is the operation that performs cuts in two different positions of A , creating four open ends, and joins these open ends in a different way [3, 24]. For example, let $A = \{\overline{61784}, 3\overline{52}\}$, and consider a DCJ that cuts between markers 1 and 7 of its first chromosome and between markers 5 and 2 of its second chromosome, creating fragments $\overline{61\bullet}$, $\bullet 784$, $3\overline{5\bullet}$ and $\bullet 2$ (where the symbols \bullet represent the open ends). If we join the first with the fourth and the third with the second open end, we get $A' = \{\overline{612}, 3\overline{5784}\}$, that is, the described DCJ operation is a translocation transforming A into A' . Indeed, a DCJ operation can correspond not only to a translocation but to several structural rearrangements, such as an inversion, a fusion or a fission.
2. **Indel operations modify the content of a genome:** We can modify the content of a genome with *insertions* and with *deletions* of blocks of contiguous markers, collectively called *indel* operations [6, 25]. As an example, consider the deletion of fragment 78 from chromosome $\overline{61784}$, resulting in chromosome $\overline{614}$. In the model we consider, we do not allow that a marker is deleted and reinserted, nor inserted and then deleted. Furthermore, at most one chromosome can be entirely deleted or inserted at once.

Let A and B be two genomes and let \mathcal{A} be the set of markers in genome A and \mathcal{B} be the set of markers in genome B . We consider two distinct settings:

- In a *family-based setting* markers are grouped into families and each marker from a genome is represented by its family. Therefore, a marker from \mathcal{A} can occur more than once in A , as well as a marker from \mathcal{B} can occur more than once in B . Furthermore, genomes A and B may share a set of *common markers* $\mathcal{G} = \mathcal{A} \cap \mathcal{B}$. We also have sets $\mathcal{A}_* = \mathcal{A} \setminus \mathcal{G}$ and $\mathcal{B}_* = \mathcal{B} \setminus \mathcal{G}$ of markers that occur respectively only in A and only in B and are called *exclusive markers*. For example, consider genomes $A = \{\overline{31432}, 3\overline{52}\}$ and $B = \{\overline{121326}\}$. In this case we have $\mathcal{A} = \{1, 2, 3, 4, 5\}$ and $\mathcal{B} = \{1, 2, 3, 6\}$. Consequently, $\mathcal{G} = \{1, 2, 3\}$, $\mathcal{A}_* = \{4, 5\}$ and $\mathcal{B}_* = \{6\}$.
- In a *family-free setting* the markers of A and B are all distinct and unique. In other words, each marker of \mathcal{A} occurs exactly once in A , each marker of \mathcal{B} occurs exactly once in B and $\mathcal{A} \cap \mathcal{B} = \emptyset$. Consider, for example, genomes $A = \{\overline{1342}\}$ and $B = \{\overline{875}, 9\overline{6}\}$.

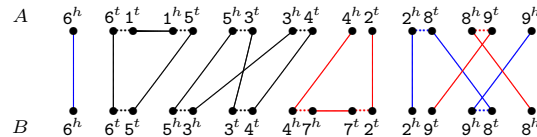
2.1 Relational diagram and distance of family-based singular genomes

Let A and B be two genomes in a family-based setting and assume that both A and B are *singular*, that is, each marker from $\mathcal{G} = \mathcal{A} \cap \mathcal{B}$ occurs exactly once in each genome. We will now describe how the DCJ-indel distance can be computed in this case [6].

For a given marker m , denote its two extremities by m^t (tail) and m^h (head). Given two singular genomes A and B , the *relational diagram* $R(A, B)$ [4] has a set of vertices $V = V(A) \cup V(B)$, where $V(A)$ is the set of extremities of markers from A and $V(B)$ is the set of extremities of markers from B . There are three types of edges in $R(A, B)$:

- *Adjacency edges*: for each pair of marker extremities γ_1 and γ_2 that are adjacent in a chromosome of any of the two genomes, we have the adjacency edge $\gamma_1\gamma_2$. Denote by E_{adj}^A and by E_{adj}^B the adjacency edges in A and in B , respectively. Marker extremities located at chromosome ends are called *telomeres* and are not connected to any adjacency edge.
- *Extremity edges*, whose set is denoted by E_γ : for each common marker $m \in \mathcal{G}$, we have two extremity edges, one connecting the vertex m^h from $V(A)$ to the vertex m^h from $V(B)$ and the other connecting the vertex m^t from $V(A)$ to the vertex m^t from $V(B)$.
- *Indel edges*: for each occurrence of an exclusive marker $m \in \mathcal{A}_* \cup \mathcal{B}_*$, we have the indel edge $m^t m^h$. Denote by E_{id}^A and by E_{id}^B the indel edges in A and in B .

Each vertex has degree one or two: it is connected either to an extremity edge or to an indel edge, and to at most one adjacency edge, therefore $R(A, B)$ is a simple collection of cycles and paths. A path that has one endpoint in genome A and the other in genome B is called an *AB-path*. In the same way, both endpoints of an *AA-path* are in A and both endpoints of a *BB-path* are in B . A cycle contains either zero or an even number of extremity edges. When a cycle has at least two extremity edges, it is called an *AB-cycle*. Moreover, a path (respectively cycle) of $R(A, B)$ composed exclusively of indel and adjacency edges in one of the two genomes corresponds to a whole linear (respectively circular) chromosome and is called a *linear* (respectively *circular*) *singleton* in that genome. Actually, linear singletons are particular cases of *AA-* or *BB-*paths. The numbers of telomeres and of *AB-*paths in $R(A, B)$ are even. An example of a relational diagram is given in Figure 1.



■ **Figure 1** For genomes $A = \{\bar{6}1534, 289\}$ and $B = \{\bar{6}5\bar{3}4\bar{7}2, 98\}$, the relational diagram contains two cycles, two *AB-*paths (represented in blue), one *AA-*path and one *BB-*path (both represented in red). Short dotted horizontal edges are adjacency edges, long horizontal edges are indel edges, top-down edges are extremity edges.

DCJ distance of canonical genomes. When singular genomes A and B have no exclusive markers, that is, $\mathcal{A}_* = \mathcal{B}_* = \emptyset$, they are said to be *canonical*. In this case A can be sorted into B with DCJ operations only and their DCJ distance d_{DCJ} can be computed as follows [3]:

$$d_{\text{DCJ}}(A, B) = |\mathcal{G}| - c - \frac{i}{2},$$

where c is the number of *AB-*cycles and i is the number of *AB-*paths in $R(A, B)$.

Runs and indel-potential. When singular genomes A and B have exclusive markers, it is possible to optimally select DCJ operations that group exclusive markers together for minimizing indels [6], as follows.

Given two genomes A and B and a component C of $R(A, B)$, a *run* [6] is a maximal subpath of C , in which the first and the last edges are indel edges, and all indel edges belong to the same genome. It can be an \mathcal{A} -run when its indel edges are in genome A , or a \mathcal{B} -run when its indel edges are in genome B . We denote by $\Lambda(C)$ the number of runs in component C . If $\Lambda(C) \geq 1$ the component C is said to be *indel-enclosing*, otherwise $\Lambda(C) = 0$ and C is said to be *indel-free*. The *indel-potential* of a component C , denoted by $\lambda(C)$, is the optimal number of indels obtained after “sorting” C separately and can be directly computed from $\Lambda(C)$ [6]:

$$\lambda(C) = \begin{cases} 0, & \text{if } \Lambda(C) = 0 \text{ (} C \text{ is indel-free);} \\ \lceil \frac{\Lambda(C)+1}{2} \rceil, & \text{if } \Lambda(C) \geq 1 \text{ (} C \text{ is indel-enclosing).} \end{cases}$$

With the indel-potential, an upper bound for the DCJ-indel distance $d_{\text{DCJ}}^{\text{ID}}$ was established [6]:

$$d_{\text{DCJ}}^{\text{ID}}(A, B) \leq |\mathcal{G}| - c - \frac{i}{2} + \sum_{C \in R(A, B)} \lambda(C) \quad (1)$$

DCJ-indel distance of singular circular genomes. For singular circular genomes, the graph $R(A, B)$ is composed of cycles only. In this case the upper bound given by Equation (1) is tight and leads to a simplified formula [6]:

$$d_{\text{DCJ}}^{\text{ID}}(A, B) = |\mathcal{G}| - c + \sum_{C \in R(A, B)} \lambda(C).$$

DCJ-indel distance of singular linear genomes. For singular linear genomes, the upper bound given by Equation (1) is achieved when the components of $R(A, B)$ are sorted separately. However, it can be decreased by *recombinations*, that are DCJ operations that act on two distinct paths of $R(A, B)$. Such path recombinations are said to be *deducting*. The total number of types of deducting recombinations is relatively small. By exhaustively exploring the space of recombination types, it is possible to identify groups of chained recombinations [6], so that the sources of each group are the original paths of the graph. In other words, a path that is a resultant of a group is never a source of another group. This results in a greedy approach (detailed in [6]) that optimally finds the value $\delta \geq 0$ to be deducted. We then have the following exact formula [6]:

$$d_{\text{DCJ}}^{\text{ID}}(A, B) = |\mathcal{G}| - c - \frac{i}{2} + \sum_{C \in R(A, B)} \lambda(C) - \delta.$$

3 The family-free setting

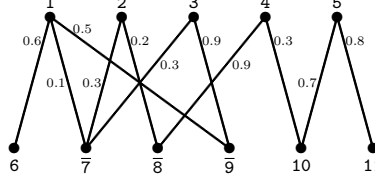
As already stated, in the family-free setting, each marker in each genome is represented by a distinct symbol, thus $\mathcal{A} \cap \mathcal{B} = \emptyset$. Observe that the cardinalities $|\mathcal{A}|$ and $|\mathcal{B}|$ may be distinct.

3.1 Marker similarity graph for the family-free setting

Given a threshold $0 \leq x \leq 1$, we can represent the similarities between the markers of genome A and the markers of genome B in the so called *marker similarity graph* [5], denoted by

3:6 Natural Family-Free Genomic Distance

$\mathcal{S}_x(A, B)$. This is a weighted bipartite graph whose partitions \mathcal{A} and \mathcal{B} are the sets of markers in genomes A and B , respectively. Furthermore, for each pair of markers $a \in \mathcal{A}$ and $b \in \mathcal{B}$, denote by $\sigma(a, b)$ their *normalized similarity*, a value that ranges in the interval $[0, 1]$. If $\sigma(a, b) \geq x$ there is an edge e connecting a and b in $\mathcal{S}_x(A, B)$ whose weight is $\sigma(e) := \sigma(a, b)$. An example is given in Figure 2.



■ **Figure 2** Graph $\mathcal{S}_{0.1}(A, B)$ for the two genomes $A = \{1\ 2\ 3\ 4\ 5\}$ and $B = \{6\ \bar{7}\ \bar{8}\ \bar{9}\ 10\ 11\}$.

Mapped genomes. Let A and B be two genomes with marker similarity graph $\mathcal{S}_x(A, B)$ and let $M = \{e_1, e_2, \dots, e_n\}$ be a matching in $\mathcal{S}_x(A, B)$. 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 $s(a, M) = s(b, M) = i$. We then define the set of *M-saturated mapped markers* $\mathcal{G}(M) = \{s(g, M) : g \text{ is } M\text{-saturated}\} = \{1, 2, \dots, n\}$.

Let \tilde{n}_A be the number of unsaturated markers in \mathcal{A} and \tilde{n}_B be the number of unsaturated markers in \mathcal{B} . We extend the function s , so that it maps each unsaturated marker $a' \in \mathcal{A}$ to one value in $\{n + 1, n + 2, \dots, n + \tilde{n}_A\}$ and each unsaturated marker $b' \in \mathcal{B}$ to one value in $\{n + \tilde{n}_A + 1, n + \tilde{n}_A + 2, \dots, n + \tilde{n}_A + \tilde{n}_B\}$. The sets of *M-unsaturated mapped markers* are:

- $\mathcal{A}_*(M) = \{s(a', M) : a' \in \mathcal{A} \text{ is } M\text{-unsaturated}\} = \{n + 1, n + 2, \dots, n + \tilde{n}_A\}$ and
- $\mathcal{B}_*(M) = \{s(b', M) : b' \in \mathcal{B} \text{ is } M\text{-unsaturated}\} = \{n + \tilde{n}_A + 1, n + \tilde{n}_A + 2, \dots, n + \tilde{n}_A + \tilde{n}_B\}$.

The *mapped genomes* A^M and B^M are then obtained by renaming each marker $a \in \mathcal{A}$ to $s(a, M)$ and each marker $b \in \mathcal{B}$ to $s(b, M)$, preserving all orientations.

Established distances of mapped genomes. Let the relational graph $R(A^M, B^M)$ have c_M AB -cycles and i_M AB -paths. By simply ignoring the exclusive markers of $\mathcal{A}_*(M)$ and $\mathcal{B}_*(M)$, we can compute the DCJ distance:

$$d_{\text{DCJ}}(A^M, B^M) = |M| - c_M - \frac{i_M}{2}.$$

Taking into consideration the weight of the matching M defined as $w(M) = \sum_{e \in M} \sigma(e)$, we can also compute the weighted DCJ distance $\text{wd}_{\text{DCJ}}(A^M, B^M)$ [16]:

$$\text{wd}_{\text{DCJ}}(A^M, B^M) = d_{\text{DCJ}}(A^M, B^M) + |M| - w(M).$$

Observe that, when all edges of M have the maximum weight 1, we have $w(M) = |M|$ and $\text{wd}_{\text{DCJ}}(A^M, B^M) = d_{\text{DCJ}}(A^M, B^M)$.

Finally, taking into consideration the exclusive markers of $\mathcal{A}_*(M)$ and $\mathcal{B}_*(M)$, but not the weight $w(M)$, we can compute the DCJ-indel distance of mapped genomes A^M and B^M :

$$d_{\text{DCJ}}^{\text{ID}}(A^M, B^M) = |M| - c_M - \frac{i_M}{2} + \sum_{C \in R(A^M, B^M)} \lambda(C) - \delta_M,$$

where δ_M is the deduction given by path recombinations in $R(A^M, B^M)$.

3.2 The family-free DCJ-indel distance

Let A^M and B^M be the mapped genomes for a given matching M of $\mathcal{S}_x(A, B)$. The *weighted relational diagram* of A^M and B^M , denoted by $WR(A^M, B^M)$, is obtained by constructing the relational diagram of A^M and B^M and adding weights to the indel edges as follows. For each mapped M -unsaturated marker $m \in \mathcal{A}_*(M) \cup \mathcal{B}_*(M)$, the indel edge $m^h m^t$ receives a weight $w(m^h m^t) = \max\{\sigma(uv) \mid uv \in \mathcal{S}_x(A, B) \text{ and } u = s^{-1}(m, M)\}$, that is the maximum similarity among the edges incident to the marker $u = s^{-1}(m, M)$ in $\mathcal{S}_x(A, B)$. We denote by $\widetilde{M} = E_{\text{id}}^A \cup E_{\text{id}}^B$ the set of indel edges, here also called the *complement* of M . The weight of \widetilde{M} is $w(\widetilde{M}) = \sum_{e \in \widetilde{M}} w(e)$. Examples of diagrams of mapped genomes are shown in Figure 3.

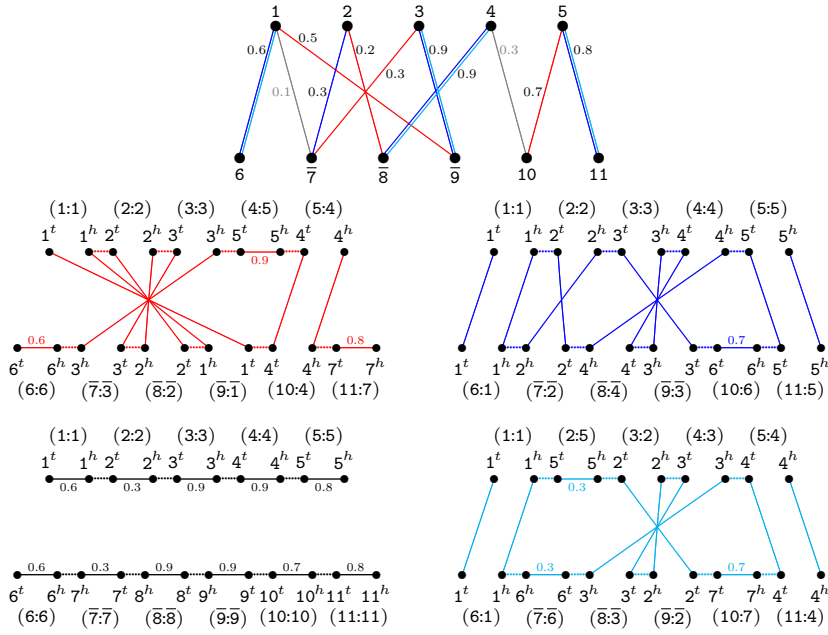


Figure 3 Considering the same genomes $A = \{1\ 2\ 3\ 4\ 5\}$ and $B = \{6\ \bar{7}\ \bar{8}\ \bar{9}\ 10\ 11\}$ as in Figure 2, let M_1 (red) and M_2 (blue) be two distinct maximal matchings in $\mathcal{S}_{0,1}(A, B)$. We also represent the non-maximal matching M_3 (cyan) that is a subset of M_2 . In the middle part we show diagrams $WR(A^{M_1}, B^{M_1})$ and $WR(A^{M_2}, B^{M_2})$, both with two AB -paths and two AB -cycles. In the lower part we show diagrams $WR(A^{M_\emptyset}, B^{M_\emptyset})$, corresponding to the trivial empty matching M_\emptyset and with two linear singletons (one AA -path and one BB -path), and $WR(A^{M_3}, B^{M_3})$, with two AB -paths and two AB -cycles. The labeling $(X:Y)$ indicates that $Y = s(X, M_i)$.

In the computation of the weighted DCJ-indel distance of mapped genomes A^M and B^M , denoted by $\text{wd}_{\text{DCJ}}^{\text{id}}(A^M, B^M)$, we should take into consideration the exclusive markers of $\mathcal{A}_*(M)$ and $\mathcal{B}_*(M)$, and the weights $w(M)$ and $w(\widetilde{M})$. An important condition is that $\text{wd}_{\text{DCJ}}^{\text{id}}(A^M, B^M)$ must be equal to $d_{\text{DCJ}}^{\text{id}}(A^M, B^M)$ if $w(M) = |M|$ and $w(\widetilde{M}) = 0$. We can achieve this by extending the formula for computing $\text{wd}_{\text{DCJ}}(A^M, B^M)$ as follows:

$$\begin{aligned} \text{wd}_{\text{DCJ}}^{\text{id}}(A^M, B^M) &= \text{wd}_{\text{DCJ}}(A^M, B^M) + \sum_{C \in WR(A^M, B^M)} \lambda(C) - \delta_M + w(\widetilde{M}) \\ &= d_{\text{DCJ}}(A^M, B^M) + |M| - w(M) + \sum_{C \in WR(A^M, B^M)} \lambda(C) - \delta_M + w(\widetilde{M}) \\ &= d_{\text{DCJ}}^{\text{id}}(A^M, B^M) + |M| - w(M) + w(\widetilde{M}). \end{aligned}$$

Let us now examine the behaviour of the formula above for the examples given in Figure 3. Matching M_1 is maximal and gives the distance $\text{wd}_{\text{DCJ}}^{\text{ID}}(A^{M_1}, B^{M_1}) = 8.6$. Matching M_2 is also maximal and gives the distance $\text{wd}_{\text{DCJ}}^{\text{ID}}(A^{M_2}, B^{M_2}) = 5.2$. The empty matching M_\emptyset gives the distance $\text{wd}_{\text{DCJ}}^{\text{ID}}(A^{M_\emptyset}, B^{M_\emptyset}) = 9.7$, that is the biggest. And the non-maximal matching $M_3 \subset M_2$ gives the distance $\text{wd}_{\text{DCJ}}^{\text{ID}}(A^{M_3}, B^{M_3}) = 5.1$, that is the smallest.

Given that \mathbb{M} is the set of all distinct matchings in $\mathcal{S}_x(A, B)$, the family-free DCJ-indel distance is defined as follows:

$$\text{ffd}_{\text{DCJ}}^{\text{ID}}(A, B, \mathcal{S}_x) = \min_{M \in \mathbb{M}} \{\text{wd}_{\text{DCJ}}^{\text{ID}}(A^M, B^M)\}.$$

Complexity. If two family-based genomes contain the same number of occurrences of each marker, they are said to be *balanced*. The problem of computing the DCJ distance of balanced genomes (BG-DCJ) is NP-hard [23]. Since the computation of $\text{ffd}_{\text{DCJ}}^{\text{ID}}$ can be used to solve BG-DCJ, it is also NP-hard. The details of the reduction can be found in [19].

4 Family-free relational diagram

An efficient way to solve the family-free DCJ-indel distance is to develop an ILP that searches for its solution in a general graph, that represents all possible diagrams corresponding to all candidate matchings, in a similar way as the approaches given in [4, 16, 23]. Given two genomes A and B and their marker similarity graph $\mathcal{S}_x(A, B)$, the structure that integrates the properties of all diagrams of mapped genomes is the *family-free relational diagram* $\text{FFR}(A, B, \mathcal{S}_x)$, that has a set $V(A)$ with a vertex for each of the two extremities of each marker of genome A and a set $V(B)$ with a vertex for each of the two extremities of each marker of genome B .

Again, sets E_{adj}^A and E_{adj}^B contain adjacency edges connecting adjacent extremities of markers in A and in B . But here the set E_γ contains, for each edge $ab \in \mathcal{S}_x(A, B)$, an extremity edge connecting a^t to b^t , and an extremity edge connecting a^h to b^h . To both edges $a^t b^t$ and $a^h b^h$, that are called *siblings*, we assign the same weight, that corresponds to the similarity of the edge ab in $\mathcal{S}_x(A, B)$: $w(a^t b^t) = w(a^h b^h) = \sigma(ab)$. Furthermore, for each marker m there is an indel edge connecting the vertices m^h and m^t . The indel edge $m^h m^t$ receives a weight $w(m^h m^t) = \max\{\sigma(mv) | mv \in \mathcal{S}_x(A, B)\}$, that is, it is the maximum similarity among the edges incident to the marker m in $\mathcal{S}_x(A, B)$. We denote by E_{id}^A the set of indel edges of markers in genome A and by E_{id}^B the set of indel edges of markers in genome B . An example of a family-free relational diagram is given in Figure 4.

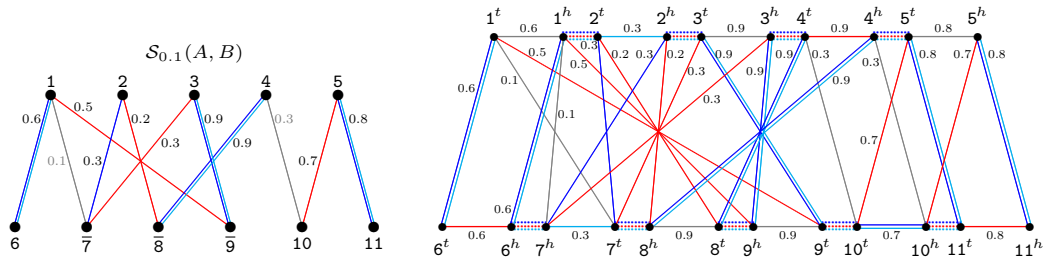


Figure 4 Given genomes $A = \{1\ 2\ 3\ 4\ 5\}$ and $B = \{6\ \bar{7}\ \bar{8}\ \bar{9}\ 10\ 11\}$, in the left part we represent the marker similarity graph $\mathcal{S}_{0.1}(A, B)$ and in the right part the family-free relational diagram $\text{FFR}(A, B, \mathcal{S}_{0.1})$. We represent in multiple colors the edges that correspond to multiple matchings.

4.1 Consistent decompositions

The diagram $FFR(A, B, \mathcal{S}_x)$ may contain vertices of degree larger than two. A *decomposition* of $FFR(A, B, \mathcal{S}_x)$ is a collection of vertex-disjoint *components*, that can be cycles and/or paths, covering all vertices of $FFR(A, B, \mathcal{S}_x)$. There can be multiple ways of selecting a decomposition, and we need to find one that allows to identify a matching of $\mathcal{S}_x(A, B)$. A set $S \subseteq E_\gamma$ is a *sibling-set* if it is exclusively composed of pairs of siblings and does not contain any pair of incident edges. Thus, a sibling-set S of $FFR(A, B, \mathcal{S}_x)$ corresponds to a matching of $\mathcal{S}_x(A, B)$. In other words, there is a clear bijection between matchings of $\mathcal{S}_x(A, B)$ and sibling-sets of $FFR(A, B, \mathcal{S}_x)$ and we denote by M_S the matching corresponding to the sibling-set S .

The set of edges $D[S]$ induced by a sibling-set S is said to be a *consistent decomposition* of $FFR(A, B, \mathcal{S}_x)$ and can be obtained as follows. In the beginning, $D[S]$ is the union of S with the sets of adjacency edges E_{adj}^A and E_{adj}^B . We then need to determine the *complement* of the sibling-set S , denoted by \tilde{S} , that is composed of the indel-edges of $FFR(A, B, \mathcal{S}_x)$ that must be added to $D[S]$: for each indel edge e , if its two endpoints have degree one or zero in $D[S]$, then e is added to both \tilde{S} and $D[S]$. (Note that $\tilde{S} = \tilde{M}_S$, while $|S| = 2|M_S|$ and $w(S) = 2w(M_S)$.) The consistent decomposition $D[S]$ covers all vertices of $FFR(A, B, \mathcal{S}_x)$ and is composed of cycles and paths, allowing us to compute the values

$$d_{\text{DCJ}}^{\text{ID}}(D[S]) = \frac{|S|}{2} - c_D - \frac{i_D}{2} + \sum_{C \in D[S]} \lambda(C) - \delta_D \quad \text{and}$$

$$\text{wd}_{\text{DCJ}}^{\text{ID}}(D[S]) = d_{\text{DCJ}}^{\text{ID}}(D[S]) + \frac{|S|}{2} - \frac{w(S)}{2} + w(\tilde{S}),$$

where c_D and i_D are the numbers of AB -cycles and AB -paths in $D[S]$, respectively, and δ_D is the optimal deduction of recombinations of paths from $D[S]$.

Given that \mathbb{S} is the sets of all sibling-sets of $FFR(A, B, \mathcal{S}_x)$, we compute the family-free DCJ-indel distance of A and B with the following equation:

$$\text{ffd}_{\text{DCJ}}^{\text{ID}}(A, B, \mathcal{S}_x) = \min_{S \in \mathbb{S}} \{ \text{wd}_{\text{DCJ}}^{\text{ID}}(D[S]) \}.$$

4.2 Capping

Telomeres produce some difficulties for the decomposition of $FFR(A, B, \mathcal{S}_x)$, and a known technique to overcome this problem is called *capping* [13]. It consists of modifying the diagram by adding *artificial* markers, also called *caps*, whose extremities should be properly connected to the telomeres of the linear chromosomes of A and B . Therefore, usually the capping depends on the numbers κ_A and κ_B , that are, respectively, the total numbers of linear chromosomes in genomes A and B .

Family-based singular genomes. First we recall the capping of family-based singular genomes. Here the caps must circularize all linear chromosomes, so that their relational diagram is composed of cycles only, but, if the capping is optimal, the DCJ-indel distance is preserved.

An optimal capping that transforms singular linear genomes A and B into singular circular genomes can be obtained after identifying the recombination groups [6]. The DCJ-indel distance is preserved by properly linking the components of each identified recombination group into a single cycle [4]. Such a capping may require some artificial adjacencies between caps. The following result is very useful.

► **Theorem 1** (from [4]). *We can obtain an optimal capping of singular genomes A and B with exactly $p_* = \max\{\kappa_A, \kappa_B\}$ caps and $|\kappa_A - \kappa_B|$ artificial adjacencies between caps.*

Capped family-free relational diagram. We transform $FFR(A, B, \mathcal{S}_x)$ into the *capped family-free relational diagram* $FFR_o(A, B, \mathcal{S}_x)$ as follows. Again, let $p_* = \max\{\kappa_A, \kappa_B\}$. The diagram $FFR_o(A, B, \mathcal{S}_x)$ is obtained by adding to $FFR(A, B, \mathcal{S}_x)$ $4p_*$ new vertices, named $\circ_A^1, \circ_A^2, \dots, \circ_A^{2p_*}$ and $\circ_B^1, \circ_B^2, \dots, \circ_B^{2p_*}$, each one representing a *cap extremity*. Each of the $2\kappa_A$ telomeres of A is connected by an adjacency edge to a distinct cap extremity among $\circ_A^1, \circ_A^2, \dots, \circ_A^{2\kappa_A}$. Similarly, each of the $2\kappa_B$ telomeres of B is connected by an adjacency edge to a distinct cap extremity among $\circ_B^1, \circ_B^2, \dots, \circ_B^{2\kappa_B}$. Moreover, if $\kappa_A < \kappa_B$, for $i = 2\kappa_A + 1, 2\kappa_A + 3, \dots, 2\kappa_B - 1$, connect \circ_A^i to \circ_A^{i+1} by an *artificial adjacency edge*. Otherwise, if $\kappa_B < \kappa_A$, for $j = 2\kappa_B + 1, 2\kappa_B + 3, \dots, 2\kappa_A - 1$, connect \circ_B^j to \circ_B^{j+1} by an artificial adjacency edge. All these new adjacency edges and artificial adjacency edges are added to E_{adj}^A and E_{adj}^B , respectively. We also connect each \circ_A^i , $1 \leq i \leq 2p_*$, by a *cap extremity edge* to each \circ_B^j , $1 \leq j \leq 2p_*$, and denote by E_o the set of cap extremity edges.

A set $P \subseteq E_o$ is a *capping-set* if it does not contain any pair of incident edges and is maximal. Since each cap extremity of A is connected to each cap extremity of B , the size of any (maximal) capping-set is $2p_*$. A consistent decomposition $Q[S, P]$ of $FFR_o(A, B, \mathcal{S}_x)$ is induced by a sibling-set $S \subseteq E_\gamma$ and a (maximal) capping-set $P \subseteq E_o$ and is composed of vertex disjoint cycles that cover all vertices of $FFR_o(A, B, \mathcal{S}_x)$.

► **Theorem 2.** Let \mathbb{P}_{MAX} be the set of all distinct (maximal) capping-sets from $FFR_o(A, B, \mathcal{S}_x)$. For each sibling-set S of $FFR(A, B, \mathcal{S}_x)$ and $FFR_o(A, B, \mathcal{S}_x)$, we have

$$d_{\text{DCJ}}^{\text{ID}}(D[S]) = \min_{P \in \mathbb{P}_{\text{MAX}}} \{d_{\text{DCJ}}^{\text{ID}}(Q[S, P])\}, \text{ and}$$

$$\text{wd}_{\text{DCJ}}^{\text{ID}}(D[S]) = \min_{P \in \mathbb{P}_{\text{MAX}}} \{\text{wd}_{\text{DCJ}}^{\text{ID}}(Q[S, P])\}.$$

Proof. Each capping-set corresponds to exactly p_* caps. In addition, all adjacencies, including the $|\kappa_A - \kappa_B|$ artificial adjacencies between cap extremities, are part of each consistent decomposition. Recall that each sibling-set S of $FFR_o(A, B, \mathcal{S}_x)$ corresponds to a matching M_S of $\mathcal{S}_x(A, B)$. The set of consistent decompositions include all possible distinct consistent decompositions induced by S together with one distinct element of \mathbb{P}_{MAX} . Theorem 1 states that the pair of matched genomes A^{M_S} and B^{M_S} can be optimally capped with p_* caps and $|\kappa_A - \kappa_B|$ artificial adjacencies. Therefore, it is clear that $d_{\text{DCJ}}^{\text{ID}}(D[S]) = \min_{P \in \mathbb{P}_{\text{MAX}}} \{d_{\text{DCJ}}^{\text{ID}}(Q[S, P])\}$. Since the capping does not change the sizes of the sibling-sets and their weights and complements, it is also clear that $\text{wd}_{\text{DCJ}}^{\text{ID}}(D[S]) = \min_{P \in \mathbb{P}_{\text{MAX}}} \{\text{wd}_{\text{DCJ}}^{\text{ID}}(Q[S, P])\}$. ◀

Alternative formula for computing the indel-potential of cycles. The consistent decompositions of $FFR_o(A, B, \mathcal{S}_x)$ are composed exclusively of cycles, and the number of runs $\Lambda(C)$ of a cycle C is always in $\{0, 1, 2, 4, 6, \dots\}$. Therefore, the formula to compute the indel-potential of a cycle C can be simplified to

$$\lambda(C) = \begin{cases} \Lambda(C), & \text{if } \Lambda(C) \in \{0, 1\} \\ 1 + \frac{\Lambda(C)}{2}, & \text{if } \Lambda(C) \in \{2, 4, 6, \dots\} \end{cases}$$

that can still be redesigned to a form that can be easier implemented in the ILP [4]. First, let a *transition* in a cycle C be an indel-free segment of C that is between a run in one genome and a run in the other genome and denote by $\aleph(C)$ the number of transitions in C . Observe that, if C is indel-free, then obviously $\aleph(C) = 0$. If C has a single run, then we also have $\aleph(C) = 0$. On the other hand, if C has at least 2 runs, then $\aleph(C) = \Lambda(C)$. The new formula is split into two parts. The first part is the function $r(C)$, defined as $r(C) = 1$ if

$\Lambda(C) \geq 1$, otherwise $r(C) = 0$, that simply tests whether C is indel-enclosing or indel-free. The second part depends on the number of transitions $\aleph(C)$, and the complete formula stands as follows [4]:

$$\lambda(C) = r(C) + \frac{\aleph(C)}{2}.$$

Distance formula. Note that the number of indel-enclosing components is $\sum_{C \in Q[S, P]} r(C) = c_Q^r + s_Q$, where c_Q^r and s_Q are the number of indel-enclosing AB -cycles and the number of circular singletons in $Q[S, P]$, respectively. Furthermore, the number of indel-free AB -cycles of $Q[S, P]$ is $c_Q^{\tilde{r}} = c_Q - c_Q^r$. We can now compute the values

$$\begin{aligned} d_{\text{DCJ}}^{\text{ID}}(Q[S, P]) &= p_* + \frac{|S|}{2} - c_Q + \sum_{C \in Q[S, P]} \lambda(C) \\ &= p_* + \frac{|S|}{2} - c_Q + \sum_{C \in Q[S, P]} \left(r(C) + \frac{\aleph(C)}{2} \right) \\ &= p_* + \frac{|S|}{2} - c_Q^{\tilde{r}} + s_Q + \sum_{C \in Q[S, P]} \frac{\aleph(C)}{2}, \quad \text{and} \\ \text{wd}_{\text{DCJ}}^{\text{ID}}(Q[S, P]) &= d_{\text{DCJ}}^{\text{ID}}(Q[S, P]) + \frac{|S|}{2} - \frac{w(S)}{2} + w(\tilde{S}) \\ &= p_* + |S| - c_Q^{\tilde{r}} + s_Q + \sum_{C \in Q[S, P]} \frac{\aleph(C)}{2} - \frac{w(S)}{2} + w(\tilde{S}). \end{aligned} \quad (2)$$

Given that \mathbb{S} and \mathbb{P}_{MAX} are, respectively, the sets of all sibling-sets and all maximal capping-sets of $\text{FFR}_o(A, B, \mathcal{S}_x)$, the final version of our optimization problem is

$$\text{ffd}_{\text{DCJ}}^{\text{ID}}(A, B, \mathcal{S}_x) = \min_{S \in \mathbb{S}, P \in \mathbb{P}_{\text{MAX}}} \{ \text{wd}_{\text{DCJ}}^{\text{ID}}(Q[S, P]) \}.$$

5 ILP formulation to compute the family-free DCJ-indel distance

Our formulation is an adaptation of the ILP for computing the DCJ-indel distance of family-based natural genomes, by Bohnenkämper et al. [4], that is itself an extension of the ILP for computing the DCJ distance of family-based balanced genomes, by Shao et al. [23]. The main differences between our approach and the approach from [4] are the underlying graphs and the objective functions. The general idea is searching for a sibling-set, that, together with a maximal capping-set, gives an optimal consistent cycle decomposition of the capped diagram $\text{FFR}_o(A, B, \mathcal{S}_x) = (V, E)$, where the set of edges comprises all disjoint sets of distinct types: $E = E_\gamma \cup E_o \cup E_{\text{adj}}^A \cup E_{\text{adj}}^B \cup E_{\text{id}}^A \cup E_{\text{id}}^B$. While in the ILP from [4] the search space is restricted to maximal sibling-sets, in the family-free DCJ-indel distance the search space includes all sibling-sets, of any size.

In Algorithm 1 we give the formulation for computing $\text{ffd}_{\text{DCJ}}^{\text{ID}}(A, B, \mathcal{S}_x)$, distributed in three main parts. Counting indel-free cycles in the decomposition makes up the first part, depicted in constraints (C.01)–(C.06), variables and domains (D.01)–(D.03). The second part is for counting transitions, described in constraints (C.07)–(C.10), variables and domains (D.04)–(D.05). The last part describes how to count the number of circular singletons, with constraint (C.11), variable and domain (D.06). The objective function of our ILP

3:12 Natural Family-Free Genomic Distance

minimizes the size of the sibling-set, with sum over variables x_e , the number of circular singletons, calculated by the sum over variables s_k , half the overall number of transitions in indel-enclosing AB -cycles, calculated by the sum over variables t_e , and the weight of all indel edges in the decomposition, given by the sum over their weights $w_e x_e$ for all $e \in E_{\text{id}}$, while maximizing both the number of indel-free cycles, counted by the sum over variables z_i , and half of the weights of the extremity edges in the decomposition, given by the sum over their weights $w_e x_e$ for all edges $e \in E_\gamma$. The minimization is not affected by constant p_* , that is included in the objective function to keep the correspondence to Equation (2).

■ **Algorithm 1** ILP for computing the family-free DCJ-indel distance.

$$\begin{aligned}
 \min \quad & p_* + \sum_{e \in E_\gamma} x_e - \sum_{1 \leq i \leq |V|} z_i + \sum_{k \in K} s_k + \frac{1}{2} \sum_{e \in E} t_e - \frac{1}{2} \sum_{e \in E_\gamma} w_e x_e + \sum_{e \in E_{\text{id}}} w_e x_e \\
 \text{s. t.} \quad & x_e = 1 && \forall e \in E_{\text{adj}}^A \cup E_{\text{adj}}^B && \text{(C.01)} \\
 & \sum_{uv \in E} x_{uv} = 2 && \forall u \in V && \text{(C.02)} \\
 & x_e = x_d && \forall e, d \in E_\gamma, e, d \text{ are siblings} && \text{(C.03)} \\
 & \left. \begin{aligned} y_i &\leq y_j + i(1 - x_{v_i v_j}) \\ y_j &\leq y_i + j(1 - x_{v_i v_j}) \end{aligned} \right\} && \forall v_i v_j \in E && \text{(C.04)} \\
 & \left. \begin{aligned} y_i &\leq i(1 - x_{v_i v_j}) \\ y_j &\leq j(1 - x_{v_i v_j}) \end{aligned} \right\} && \forall v_i v_j \in E_{\text{id}}^A \cup E_{\text{id}}^B && \text{(C.05)} \\
 & iz_i \leq y_i && \forall 1 \leq i \leq |V| && \text{(C.06)} \\
 & \left. \begin{aligned} r_v &\leq 1 - x_{uv} \\ r_{v'} &\geq x_{u'v'} \end{aligned} \right\} && \begin{aligned} \forall uv \in E_{\text{id}}^A \\ \forall u'v' \in E_{\text{id}}^B \end{aligned} && \text{(C.07)} \\
 & \left. \begin{aligned} t_{uv} &\geq r_v - r_u - (1 - x_{uv}) \\ t_{uv} &\geq r_u - r_v - (1 - x_{uv}) \end{aligned} \right\} && \forall uv \in E && \text{(C.08)} \\
 & \sum_{d \in E_{\text{id}}^A, d \cap e \neq \emptyset} x_d - t_e \geq 0 && \forall e \in E_{\text{adj}}^A && \text{(C.09)} \\
 & t_e = 0 && \forall e \in E \setminus E_{\text{adj}}^A && \text{(C.10)} \\
 & \sum_{e \in E_{\text{id}}^k} x_e - |k| \leq s_k && \forall k \in K && \text{(C.11)} \\
 \text{and} \quad & x_e \in \{0, 1\} && \forall e \in E && \text{(D.01)} \\
 & 0 \leq y_i \leq i && \forall 1 \leq i \leq |V| && \text{(D.02)} \\
 & z_i \in \{0, 1\} && \forall 1 \leq i \leq |V| && \text{(D.03)} \\
 & r_v \in \{0, 1\} && \forall v \in V && \text{(D.04)} \\
 & t_e \in \{0, 1\} && \forall e \in E && \text{(D.05)} \\
 & s_k \in \{0, 1\} && \forall k \in K && \text{(D.06)} \\
 & p_* = \max\{\kappa_A, \kappa_B\} && && \text{(D.07)}
 \end{aligned}$$

Comparison to related models. Since the pre-requisites of a family-free setting differ substantially from those of a family-based setting, we could not compare our approach to the one from [4]. We intend to perform such a comparison in a future work, for example by using pairwise similarities to cluster the genes into families. Comparing our approach to the original family-free DCJ distance was also not possible, because the ILP provided in [16] is only suitable for unichromosomal genomes. Again, we intend to perform such a comparison in a future work, after we implement an ILP that is able to compute the family-free DCJ distance of multichromosomal genomes.

Unweighted version. In the present work, for comparison purposes, we also implemented a simpler version of the family-free DCJ-indel distance, that simply ignores all weights. This version is called *unweighted* family-free DCJ-indel distance, and consists of finding a sibling-set in $FFR_o(A, B, \mathcal{S}_x)$ that minimizes $d_{\text{DCJ}}^{\text{ID}}(D[S, P])$. But here it is important to observe that smaller sibling-sets, that simply discard blocks of contiguous markers, tend to give the smaller distances. Considering the similarity graph $\mathcal{S}_{0.1}(A, B)$ of Figure 3, the trivial empty matching gives the distance $d_{\text{DCJ}}^{\text{ID}}(A^{M_0}, B^{M_0}) = 2$ (deletion of the chromosome of A followed by the insertion of the chromosome of B). For the other matchings we have $d_{\text{DCJ}}^{\text{ID}}(A^{M_1}, B^{M_1}) = 4$ and $d_{\text{DCJ}}^{\text{ID}}(A^{M_2}, B^{M_2}) = d_{\text{DCJ}}^{\text{ID}}(A^{M_3}, B^{M_3}) = 3$. We then restrict the search space to *maximal* sibling-sets only, avoiding that blocks of markers are discarded. However, this could also enforce weak connections. In the example shown in Figure 3, both maximal matchings M_1 and M_2 have weak edges with weights 0.2 and 0.3. Matching M_3 has only edges with weight at least 0.6, but it would be ignored for being non-maximal. Enforcing weak connections can be prevented by removing them from the similarity graph, that is, by assigning a higher value to the cutting threshold x . Given that \mathcal{S}_{MAX} and \mathbb{P}_{MAX} are, respectively, the sets of all maximal sibling-sets and all maximal capping-sets of $FFR_o(A, B, \mathcal{S}_x)$, the unweighted version of the problem is then:

$$\text{unwffd}_{\text{DCJ}}^{\text{ID}}(A, B, \mathcal{S}_x) = \min_{S \in \mathcal{S}_{\text{MAX}}, P \in \mathbb{P}_{\text{MAX}}} \{d_{\text{DCJ}}^{\text{ID}}(Q[S, P])\}.$$

For computing the unweighted $\text{unwffd}_{\text{DCJ}}^{\text{ID}}(A, B, \mathcal{S}_x)$ we need to slightly modify the ILP described in Algorithm 1. The details can be found in [19].

Implementation. The ILPs for computing both the family-free DCJ-indel distance and its unweighted version were implemented and can be downloaded from our GitLab server at <https://gitlab.ub.uni-bielefeld.de/gi/gen-diff>.

Data analysis. For all pairwise comparisons, we obtained gene similarities using the FFGC pipeline² [11], with the following parameters: (i) 1 for the minimum number of genomes for which each gene must share some similarity in, (ii) 0.1 for the stringency threshold, (iii) 1 for the BLAST e-value, and (iv) default values for the remaining parameters. As an ILP solver, for all experiments we ran CPLEX with 8 2.67GHz cores.

Cutting threshold. Differently from the unweighted version, that requires a cutting threshold of about $x=0.5$ to give accurate results, the weighted $\text{ffd}_{\text{DCJ}}^{\text{ID}}$ was designed to be computed with all given pairwise similarities, i.e., with the cutting threshold $x=0$, that leads to a “complete” family-free relational diagram. Such a diagram would be too large to be handled in practice, therefore, if $x=0$, we consider only the similarities that are strictly greater than 0. Nevertheless, for bigger instances the diagram with similarities close to 0 might still be too large to be solved in reasonable time. Hence, for some instances it may be necessary to do a small increase of the cutting threshold. Our experiments in real data (described in Section 5.2) show that small similarities have a minor impact on the computed distance, therefore, by adopting a small cutting threshold x up to 0.3, it is possible to reduce the diagram and solve bigger instances, still with good accuracy.

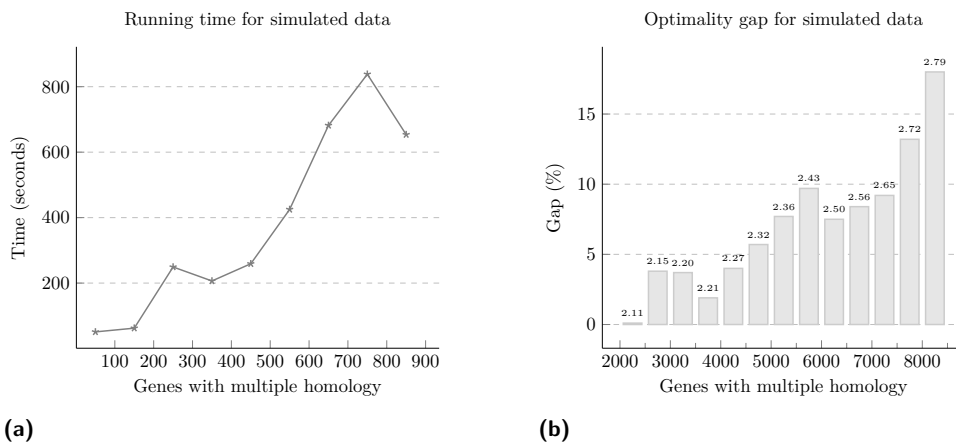
² <https://bibiserv.cebitec.uni-bielefeld.de/ffgc>

5.1 Performance evaluation

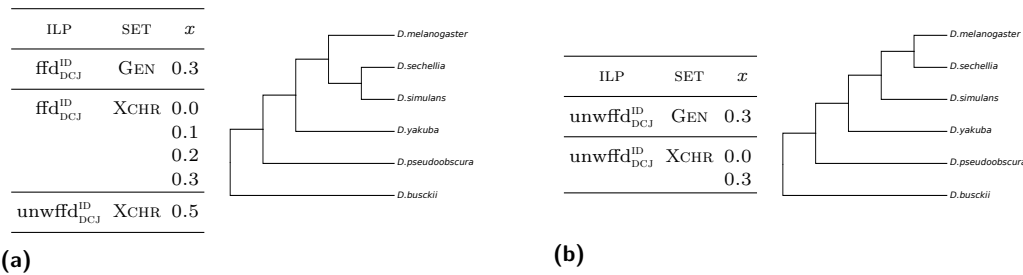
We generated simulated genomes using Artificial Life Simulator (ALF) [10] in order to benchmark our algorithm for computing the family-free DCJ-indel distance. We simulated and compared 190 pairs of genomes with different duplication rates, keeping all other parameters fixed (e.g. rearrangement, indel and mutation rates). The extant genomes have around 10,000 genes. We obtained gene similarities between simulated genomes using FFGC. For each genome pair, a threshold of $x = 0.1$ resulted in up to 8,400 genes with multiple homology relations (i.e. vertices with degree > 1 in $\mathcal{S}_{0.1}(A, B)$) and from 2 to 2.8 relations on average for those genes. In addition, each pair is about 3,000 rearrangement events away from each other. The complete parameter sets used for running ALF, together with additional information on simulated genomes, can be found in Appendix A.

For computing the family-free DCJ-indel distances, we ran CPLEX with maximum CPU time of 1 hour. Results were grouped depending on the number of genes with multiple homology relations in the respective genome pairs. Figure 5 summarizes the performance of our weighted family-free DCJ-indel distance formulation. The running times escalate quickly as the number of genes with multiple homologies increase (Figure 5a, grouped in intervals of 100), reaching the time limit after 2,000 of them (Figure 5b, grouped in intervals of 500). The optimality gap is the relative gap between the best solution found and the upper bound found by the solver, calculated by $(\frac{\text{upper bound}}{\text{best solution}} - 1) \times 100$, and appears to grow, for our simulated data, linearly in the number of genes with multiple homologies (Figure 5b).

The solution time and the optimality gap of our algorithm clearly depends less on genome sizes and more on the multiplicity of homology relations. In our experiments, we were able to find in 1 hour optimal or near-optimal solutions for genomes with 10,000 genes and up to 4,000 genes with 2.2 homology relations on average. Our formulation should be able to handle, for instance, the complete genomes of bacteria, fungi and insects, or even sets of chromosomes of mammal and plant genomes.



■ **Figure 5** Results of the weighted family-free DCJ-indel distance given by the solver, (a) shows the average running time for instances grouped by the number of vertices with degree > 1 in $\mathcal{S}_{0.1}(A, B)$ (in intervals of 100, those greater than 900 are not shown), and (b) for groups of instances that did not finish within the time limit of 1 hour, the average optimality gap and the average number of homology relations for those genes with multiple homologies (in intervals of 500).



■ **Figure 6** Based on distance matrices calculated by our ILPs for the pairwise comparisons of complete genomes (GEN) or only X chromosomes (XCHR) of *Drosophila*, we built phylogenetic trees computed by the Neighbor-Joining method [14, 20]. The output of this algorithm is an unrooted tree, and we assumed the most distant species *D. busckii* as the outgroup for rooting the trees. All comparisons converged to exactly two trees, and next to each tree we give a list of comparisons that produced that tree. The tree in (a) agrees with the reference shown in Figure 7 (Appendix B), while the tree in (b) diverges from the reference in a single branch.

5.2 Real data analysis

We evaluated the potential of our approach by comparing genomes of fruit flies from the genus *Drosophila* [1, 9, 17, 26], including the following species: *D. busckii*, *D. melanogaster*, *D. pseudoobscura*, *D. sechellia*, *D. simulans* and *D. yakuba*. A reference phylogenetic tree of these species is shown in Figure 7, in Appendix B, where we also give the sources of the DNA sequences for each analyzed genome, and additional information on the experiments. Each genome has approximately 150Mb, with about 13,000 genes distributed in 5–6 chromosomes. We obtained gene similarities using FFGC and performed two separate experiments, whose computed distances were used to build phylogenetic trees using Neighbor-Joining [14, 20].

Pairwise comparison of complete genomes. In this experiment, genomes in each comparison comprise together $\sim 13,000$ genes with multiple homologies (11.2 on average), some of them having about 90 relations considering similarities that are strictly greater than $x = 0$. Since these instances were too large, we set the threshold to $x = 0.3$. We then ran CPLEX with maximum CPU time of 3 hours. All $\text{ffd}_{\text{DCJ}}^{\text{ID}}$ computations finished within the time limit, most of them in less than 10 minutes, whereas the unweighted $\text{unwffd}_{\text{DCJ}}^{\text{ID}}$ computations, in spite of having a search space of maximal sibling-sets, that is much smaller, surprisingly took from 1 to 3 hours. We conjecture that this is due to a large number of co-optimal solutions in the unweighted version, while in $\text{ffd}_{\text{DCJ}}^{\text{ID}}$ the co-optimality is considerably minimized by weights, which helps the solver to converge faster. While the tree given by $\text{ffd}_{\text{DCJ}}^{\text{ID}}$, shown in Figure 6a, agrees with the reference tree, the tree given by $\text{unwffd}_{\text{DCJ}}^{\text{ID}}$, shown in Figure 6b, diverges from the reference in a single branch. Details of the results are given in Appendix B.1.

Pairwise comparison of X chromosomes. We also did an experiment with smaller instances, composed of pairwise comparisons of X chromosomes only, so that we could evaluate the impact of the cutting threshold on the accuracy of the approach. In this experiment, considering similarities that are strictly greater than $x = 0$, each pair comprises 1,000–2,000 genes with multiple homologies (5 on average) with as many as 30 relations.

We computed $\text{ffd}_{\text{DCJ}}^{\text{ID}}$ with cutting thresholds $x = 0$, $x = 0.1$, $x = 0.2$ and $x = 0.3$, always obtaining the accurate phylogenetic tree from Figure 6a. These results suggest that a small cutting threshold allows to reduce the size of the instances, without having a big impact in the accuracy of $\text{ffd}_{\text{DCJ}}^{\text{ID}}$.

In addition, we computed $\text{unwffid}_{\text{DCJ}}^{\text{ID}}$ with cutting thresholds $x = 0$ and $x = 0.3$, both resulting in the slightly inaccurate tree from Figure 6b, and $x = 0.5$, that also resulted in the accurate tree from Figure 6a. As expected, in the unweighted formulation the cutting threshold plays a major role in the accuracy of the calculated distances.

The analyses were done with maximum CPU time of 1 hour. The comparisons finished within a few seconds for most of instances, except for $\text{unwffid}_{\text{DCJ}}^{\text{ID}}$ with threshold $x = 0$, for which the majority of the pairwise comparisons reached the time limit – with an optimality gap of less than 3.5% though (see Appendix B.2).

Length of indel segments. As a generalization of the singular DCJ-indel model [6], the basic idea behind our approach is that runs can be merged and accumulated with DCJ operations. This is a more parsimonious alternative to the trivial approach of inserting or deleting exclusive markers individually. However, it raises the question of whether the indels then tend to be very long, and whether this makes biological sense. Considering that it is possible to distribute the runs so that each indel is composed of 1-2 runs, we can say that the lengths of the runs play a major role in defining the length of indel segments. In the particular analysis of *Drosophila* complete genomes, we have an average run length of 5.1, while the maximum run length is 121. We conjecture that the long runs are mostly composed of genes that are part of a contiguous segment from the beginning, and are not really accumulated by DCJ operations. In a future work we intend to have a closer look into the long runs, so that we can characterize their structures and verify this conjecture.

6 Conclusions and discussion

In this work we proposed a new genomic distance, for the first time integrating DCJ and indel operations in a family-free setting. In this setting the whole analysis requires less pre-processing and no classification of the data, since it can be performed based on the pairwise similarities of markers in both genomes. Based on the positions and orientations of markers in both genomes we build the *family-free relational diagram*. We then assign weights to the edges of the diagram, according to the given pairwise similarities. A *sibling-set* of edges corresponds to a set of matched markers in both genomes. Our approach transfers weights from the edges to matched and unmatched markers, so that, again for the first time, an optimal solution does not necessarily need to maximize the number of matched markers. Instead, the search space of our approach allows solutions composed of any number of matched markers. The computation of our new family-free DCJ-indel distance is NP-hard and we provide an efficient ILP formulation to solve it.

The experiments on simulated data show that our ILP can handle not only bacterial genomes, but also complete genomes of fungi and insects, or sets of chromosomes of mammals and plants. We performed a comparison study of six fruit fly genomes, using the obtained distances to reconstruct the phylogenetic tree of the six species, obtaining accurate results. This study was a first validation of the quality of our method and a more rigorous evaluation will be performed in a future work. In particular, we intend to analyze the reasons behind insertions and deletions of long segments and verify the quality of the obtained gene matchings, by comparing them to the annotated orthologies given by public databases. Furthermore, as already mentioned, we plan to compare our ILP to the one given in [4], once we manage to cluster the genes into families, and also to implement an ILP that is able to compute the family-free DCJ distance described in [16] for multichromosomal genomes, so that we can compare it to our ILP.

References

- 1 Mark D. Adams, Susan E. Celniker, Robert A. Holt, *et al.* The genome sequence of *Drosophila melanogaster*. *Science*, 287:2185–2195, 2000. doi:10.1126/science.287.5461.2185.
- 2 Sébastien Angibaud, Guillaume Fertin, Irena Rusu, Annelise Thévenin, and Stéphane Vialette. On the approximability of comparing genomes with duplicates. *Journal of Graph Algorithms and Applications*, 13(1):19–53, 2009. doi:10.7155/jgaa.00175.
- 3 Anne Bergeron, Julia Mixtacki, and Jens Stoye. A unifying view of genome rearrangements. In *Proc. of WABI*, volume 4175 of *Lecture Notes in Bioinformatics*, pages 163–173, 2006. doi:10.1007/11851561_16.
- 4 Leonard Bohnenkämper, Marília D. V. Braga, Daniel Doerr, and Jens Stoye. Computing the rearrangement distance of natural genomes. In *Proc. of RECOMB*, volume 12074 of *Lecture Notes in Bioinformatics*, pages 3–18, 2020. doi:10.1007/978-3-030-45257-5_1.
- 5 Marília D. V. Braga, Cedric Chauve, Daniel Doerr, Katharina Jahn, Jens Stoye, Annelise Thévenin, and Roland Wittler. The potential of family-free genome comparison. In C. Chauve, N. El-Mabrouk, and E. Tannier, editors, *Models and Algorithms for Genome Evolution*, chapter 13, pages 287–307. Springer, 2013. doi:10.1007/978-1-4471-5298-9_13.
- 6 Marília D. V. Braga, Eyla Willing, and Jens Stoye. Double cut and join with insertions and deletions. *Journal of Computational Biology*, 18(9):1167–1184, 2011. doi:10.1089/cmb.2011.0118.
- 7 David Bryant. The complexity of calculating exemplar distances. In David Sankoff and Joseph H. Nadeau, editors, *Comparative Genomics*, pages 207–211. Springer, 2000. doi:10.1007/978-94-011-4309-7_19.
- 8 Laurent Bulteau and Minghui Jiang. Inapproximability of (1,2)-exemplar distance. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10(6):1384–1390, 2013. doi:10.1109/TCBB.2012.144.
- 9 Andrew G. Clark, Michael B. Eisen, Douglas R. Smith, *et al.* Evolution of genes and genomes on the *Drosophila* phylogeny. *Nature*, 450:203–218, 2007. doi:10.1038/nature06341.
- 10 Daniel A. Dalquen, Maria Anisimova, Gaston H. Gonnet, and Christophe Dessimoz. ALF – a simulation framework for genome evolution. *Mol Biol Evol*, 29(4):1115, 2012. doi:10.1093/molbev/msr268.
- 11 Daniel Doerr, Pedro Feijão, and Jens Stoye. Family-free genome comparison. In João C. Setubal, Jens Stoye, and Peter F. Stadler, editors, *Comparative Genomics: Methods and Protocols*, pages 331–342. Springer, 2018. doi:10.1007/978-1-4939-7463-4_12.
- 12 Daniel Doerr, Annelise Thévenin, and Jens Stoye. Gene family assignment-free comparative genomics. *BMC Bioinformatics*, 13(Suppl 19):S3, 2012. doi:10.1186/1471-2105-13-S19-S3.
- 13 Sridhar Hannenhalli and Pavel A. Pevzner. Transforming men into mice (polynomial algorithm for genomic distance problem). In *Proc. of FOCS*, pages 581–592, 1995. doi:10.1109/SFCS.1995.492588.
- 14 Sudhir Kumar, Glen Stecher, Michael Li, Christina Knyaz, and Koichiro Tamura. MEGA X: molecular evolutionary genetics analysis across computing platforms. *Molecular Biology and Evolution*, 35(6):1547–1549, 2018. doi:10.1093/molbev/msy096.
- 15 Sudhir Kumar, Glen Stecher, Michael Suleski, and S. Blair Hedges. Timetree: a resource for timelines, timetrees, and divergence times. *Molecular Biology and Evolution*, 34(7):1812–1819, 2017. doi:10.1093/molbev/msx116.
- 16 Fábio V. Martinez, Pedro Feijão, Marília D. V. Braga, and Jens Stoye. On the family-free DCJ distance and similarity. *Algorithms for Molecular Biology*, 13(10), 2015. doi:10.1186/s13015-015-0041-9.
- 17 Stephen Richards, Yue Liu, Brian R. Bettencourt *et al.* Comparative genome sequencing of *Drosophila pseudoobscura*: Chromosomal, gene, and cis-element evolution. *Genome Research*, 15:1–18, 2005. doi:10.1101/gr.3059305.

- 18 Diego P. Rubert, Pedro Feijão, Marília D. V. Braga, Jens Stoye, and Fábio V. Martinez. Approximating the DCJ distance of balanced genomes in linear time. *Algorithms for Molecular Biology*, 12(3), 2017. doi:10.1186/s13015-017-0095-y.
- 19 Diego P. Rubert, Fábio V. Martinez, and Marília D. V. Braga. Natural family-free genomic distance. arXiv:2007.03556, 2020.
- 20 Naruya Saitou and Masatoshi Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, 1987. doi:10.1093/oxfordjournals.molbev.a040454.
- 21 David Sankoff. Edit distance for genome comparison based on non-local operations. In *Proc. of CPM*, volume 644 of *Lecture Notes in Computer Science*, pages 121–135, 1992. doi:10.1007/3-540-56024-6_10.
- 22 David Sankoff. Genome rearrangement with gene families. *Bioinformatics*, 15(11):909–917, 1999. doi:10.1093/bioinformatics/15.11.909.
- 23 Mingfu Shao, Yu Lin, and Bernard Moret. An exact algorithm to compute the double-cut-and-join distance for genomes with duplicate genes. *Journal of Computational Biology*, 22(5):425–435, 2015. doi:10.1089/cmb.2014.0096.
- 24 Sophia Yancopoulos, Oliver Attie, and Richard Friedberg. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics*, 21(16):3340–3346, 2005. doi:10.1093/bioinformatics/bti535.
- 25 Sophia Yancopoulos and Richard Friedberg. DCJ path formulation for genome transformations which include insertions, deletions, and duplications. *Journal of Computational Biology*, 16(10):1311–1338, 2009. doi:10.1089/cmb.2009.0092.
- 26 Qi Zhou and Doris Bachtrog. Ancestral chromatin configuration constrains chromatin evolution on differentiating sex chromosomes in *Drosophila*. *PLoS Genetics*, 11(6), 2015. doi:10.1371/journal.pgen.1005331.

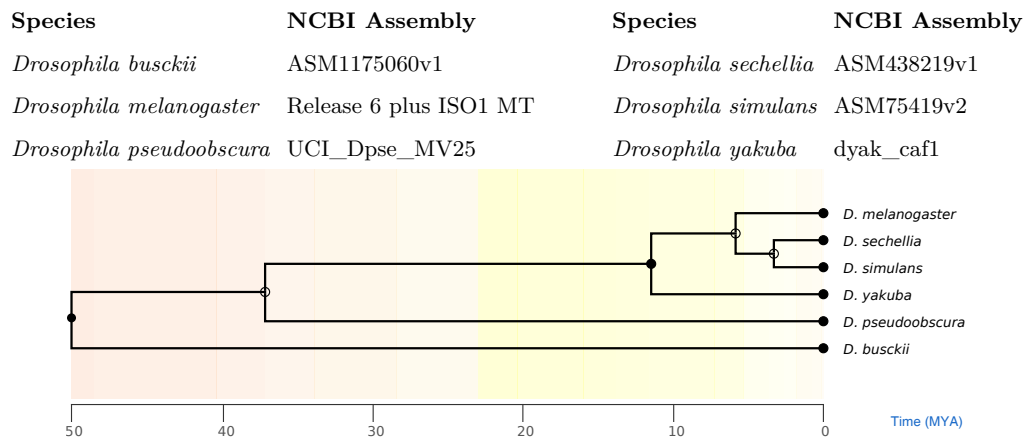
A Generation of simulated data

Here we describe the process and the parameters used in Artificial Life Simulator (ALF) [10] for generating our simulated data. Each one of the 190 instances generated consists of a pair of simulated genomes. We used the default values for parameters not mentioned. PAM units were used as time scale for simulation, starting with a randomly generated root genome with 10,000 genes, whose lengths were drawn from a Gamma distribution with $k = 2.4019$ and $\theta = 133.8063$ (minimum length 100). We used a custom evolutionary tree defining a speciation event after 25 time units, resulting in two leaf species, which evolved for additional 25 time units. The WAG substitution model was used together with Zipfian indels in DNA sequences with rate 0.0002 (maximum length 50). Such rate varies among sites according to a Gamma distribution with shape 1 and 10 classes. In addition, we set the rate of invariable sites to 0.001. Inversions and translocations of up to 30 genes were allowed at a rate of 0.0025. Finally, for generating instances comprising genes with different numbers of homologies, we varied the gene duplication and the gene loss rates between 1×10^{-5} and 2×10^{-3} .

B Analysis of *Drosophila* genomes

We downloaded the genomes of 6 species of *Drosophila* [1, 9, 17, 26] from NCBI³. In our experiments we used the assemblies listed in Figure 7, with their respective gene annotations.

³ <https://www.ncbi.nlm.nih.gov>



■ **Figure 7** List of genomes used in our experiments and a reference phylogenetic tree of the respective species of *Drosophila* given by TimeTree [15], a public knowledge-base for information on the tree-of-life and its evolutionary timescale.

As already mentioned, we obtained pairwise similarities between genes of *Drosophila* genomes using the FFGC pipeline⁴ [11] with the following parameters: (i) 1 for the minimum number of genomes for which each gene must share some similarity in, (ii) 0.1 for the stringency threshold, (iii) 1 for the BLAST e-value, and (iv) default values for the remaining parameters.

In the following subsections, in-depth information is provided on the results for experiments using complete genomes and X chromosomes of the listed *Drosophila* species.

B.1 Complete genomes

The first tables in this section detail the results of the comparison of complete genomes in terms of the BLAST alignment performed for all genes, and the corresponding similarity graphs for each genome pair without cutting threshold. This data was generated using the FFGC pipeline with the parameters described above. Unplaced scaffolds were discarded, decreasing the number of genes from $\sim 15,000$ to $\sim 13,000$. Table 1 outlines the number of gene pairs in each similarity range for each pair of genomes. Table 2 shows the number of genes with no homology relations (which induce trivial selections of indel edges in the relational diagram), the number of genes with exactly one homology relation and the number of genes with multiple homologies (which pose a significant challenge to the solver). The computed distances and elapsed time (or gap in % when the solver reaches the time limit) in the pairwise comparisons with cutting threshold 0.3 are shown in Tables 3 and 4. The solver was set to stop after finding a solution with optimality gap smaller than 0.5% or after 3 hours.

⁴ <https://bibiserv.cebitec.uni-bielefeld.de/ffgc>

■ **Table 2** Association between genes in pairwise comparisons of complete genomes, considering pairwise similarities strictly greater than 0. The tables show the number of genes with zero, one and multiple homology relations, respectively. For all of them, the element stored in line i and column j represents the number of genes of the species i in the pairwise comparison of genomes i and j .

Number of unassociated genes

species	<i>melanog</i>	<i>pseudoob</i>	<i>sechellia</i>	<i>simulans</i>	<i>yakuba</i>	<i>busckii</i>	
#genes	13049	13399	13037	13023	12835	11371	
<i>melanogaster</i>	13049	—	570	213	277	352	1183
<i>pseudoobscura</i>	13399	565	—	583	694	710	1211
<i>sechellia</i>	13037	189	620	—	263	393	1189
<i>simulans</i>	13023	335	779	345	—	484	1358
<i>yakuba</i>	12835	306	666	323	327	—	1225
<i>busckii</i>	11371	304	354	321	380	400	—

Number of genes uniquely associated

species	<i>melanog</i>	<i>pseudoob</i>	<i>sechellia</i>	<i>simulans</i>	<i>yakuba</i>	<i>busckii</i>	
#genes	13049	13399	13037	13023	12835	11371	
<i>melanogaster</i>	13049	—	5439	6624	6533	6361	5107
<i>pseudoobscura</i>	13399	5775	—	5746	5704	5707	5205
<i>sechellia</i>	13037	6650	5487	—	6656	6307	5099
<i>simulans</i>	13023	6516	5394	6594	—	6237	4985
<i>yakuba</i>	12835	6288	5358	6242	6251	—	4982
<i>busckii</i>	11371	4797	4654	4749	4730	4725	—

Number of genes associated to at least two other genes

species	<i>melanog</i>	<i>pseudoob</i>	<i>sechellia</i>	<i>simulans</i>	<i>yakuba</i>	<i>busckii</i>	
#genes	13049	13399	13037	13023	12835	11371	
<i>melanogaster</i>	13049	—	7040	6212	6239	6336	6759
<i>pseudoobscura</i>	13399	7059	—	7070	7001	6982	6983
<i>sechellia</i>	13037	6198	6930	—	6118	6337	6749
<i>simulans</i>	13023	6172	6850	6084	—	6302	6680
<i>yakuba</i>	12835	6241	6811	6270	6257	—	6628
<i>busckii</i>	11371	6270	6363	6301	6261	6246	—

■ **Table 3** Computed $\text{ffd}_{\text{DCJ}}^{\text{ID}}$ and elapsed time (or gap in %) in pairwise comparisons of complete genomes, with cutting threshold $x = 0.3$. The time limit for execution of the ILP solver is 10800s.

species	<i>pseudoobscura</i>	<i>sechellia</i>	<i>simulans</i>	<i>yakuba</i>	<i>busckii</i>
<i>melanogaster</i>	7373.7 (0.76%)	1925.5 (4431.78s)	2094.7 (109.60s)	3193.2 (201.49s)	7764.6 (540.19s)
<i>pseudoobscura</i>		7326.0 (163.12s)	7355.5 (764.24s)	7351.2 (5782.73s)	7784.0 (290.12s)
<i>sechellia</i>			1661.0 (103.33s)	3259.0 (146.88s)	7710.4 (415.23s)
<i>simulans</i>				3306.0 (216.77s)	7699.9 (115.54s)
<i>yakuba</i>					7667.4 (153.36s)

■ **Table 4** Computed $\text{unwffd}_{\text{DCJ}}^{\text{ID}}$ and elapsed time (or gap in %) in pairwise comparisons of complete genomes, with cutting threshold $x = 0.3$. The time limit for execution of the ILP solver is 10800s.

species	<i>pseudoobscura</i>	<i>sechellia</i>	<i>simulans</i>	<i>yakuba</i>	<i>busckii</i>
<i>melanogaster</i>	4084 (2.67%)	708 (0.96%)	933 (0.62%)	1269 (1.35%)	4791 (0.95%)
<i>pseudoobscura</i>		4088 (1.50%)	4176 (1.47%)	4142 (1.22%)	4797 (1.20%)
<i>sechellia</i>			905 (2812.89s)	1341 (1.10%)	4817 (0.98%)
<i>simulans</i>				1478 (1.44%)	4866 (0.84%)
<i>yakuba</i>					4820 (1.00%)

B.2 X chromosomes

Similarity values in pairwise comparisons are given in Table 5. Results for $\text{unwffd}_{\text{DCJ}}^{\text{ID}}$ and for $\text{ffd}_{\text{DCJ}}^{\text{ID}}$ are shown in Tables 6 and 7 (CPLEX was set to stop after finding a solution with optimality gap smaller than 0.1% or after 1 hour). The number of genes with 0, 1 and multiple homologies are given in Table 8.

■ **Table 5** Distribution of similarities (and percentage) in pairwise comparisons of X chromosomes.

species	similarity	<i>pseudoobscura</i>	<i>sechellia</i>	<i>simulans</i>	<i>yakuba</i>	<i>busckii</i>
<i>melanogaster</i>	(0.0-0.2)	4710 (63.70%)	829 (22.38%)	897 (25.63%)	987 (28.81%)	2072 (48.64%)
	[0.2-0.4)	980 (13.25%)	576 (15.55%)	536 (15.31%)	528 (15.41%)	738 (17.23%)
	[0.4-0.6)	541 (7.32%)	352 (9.50%)	256 (7.31%)	242 (7.06%)	475 (11.15%)
	[0.6-0.8)	605 (8.18%)	271 (7.32%)	256 (7.31%)	412 (12.03%)	584 (13.71%)
	[0.8-1.0]	558 (7.55%)	1676 (45.25%)	1555 (44.43%)	1257 (36.69%)	391 (9.18%)
		7394 (100%)	3704 (100%)	3500 (100%)	3426 (100%)	4260 (100%)
<i>pseudoobscura</i>	(0.0-0.2)		4849 (64.60%)	4703 (65.15%)	4588 (64.64%)	5021 (66.59%)
	[0.2-0.4)		962 (12.82%)	907 (12.56%)	898 (12.65%)	953 (12.64%)
	[0.4-0.6)		539 (7.18%)	498 (6.90%)	495 (6.97%)	563 (7.47%)
	[0.6-0.8)		600 (7.99%)	585 (8.10%)	574 (8.09%)	584 (7.75%)
	[0.8-1.0]		556 (7.41%)	526 (7.29%)	543 (7.65%)	419 (5.56%)
		7506 (100%)	7219 (100%)	7098 (100%)	7540 (100%)	7540 (100%)
<i>sechellia</i>	(0.0-0.2)			773 (22.62%)	961 (28.16%)	2014 (47.90%)
	[0.2-0.4)			521 (15.24%)	532 (15.59%)	741 (17.62%)
	[0.4-0.6)			191 (5.59%)	266 (7.79%)	486 (11.56%)
	[0.6-0.8)			139 (4.07%)	423 (12.39%)	574 (13.65%)
	[0.8-1.0]			1795 (52.50%)	1231 (36.07%)	390 (9.27%)
			3419 (100%)	3413 (100%)	4205 (100%)	4205 (100%)
<i>simulans</i>	(0.0-0.2)				1038 (30.77%)	2069 (49.95%)
	[0.2-0.4)				506 (15.00%)	697 (16.83%)
	[0.4-0.6)				254 (7.53%)	448 (10.82%)
	[0.6-0.8)				403 (11.95%)	556 (13.42%)
	[0.8-1.0]				1172 (34.75%)	372 (8.98%)
				3373 (100%)	4142 (100%)	4142 (100%)
<i>yakuba</i>	(0.0-0.2)					2110 (50.62%)
	[0.2-0.4)					668 (16.03%)
	[0.4-0.6)					456 (10.94%)
	[0.6-0.8)					561 (13.46%)
	[0.8-1.0]					373 (8.95%)
					4168 (100%)	4168 (100%)

■ **Table 6** Computed $\text{unwffd}_{\text{DCJ}}^{\text{ID}}$ and elapsed time (or gap in %) in pairwise comparisons of X chromosomes, with cutting thresholds $x = 0.0$, $x = 0.3$ and $x = 0.5$. The time limit is 3600s.

species	x	<i>pseudoobscura</i>	<i>sechellia</i>	<i>simulans</i>	<i>yakuba</i>	<i>busckii</i>
<i>melanogaster</i>	0.0	720 (2.80%)	132 (0.38%)	178 (30.93s)	218 (4.99s)	832 (3.49%)
	0.3	829 (1.70s)	160 (9.31s)	218 (0.98s)	293 (0.66s)	972 (0.90s)
	0.5	940 (0.46s)	228 (0.52s)	298 (0.40s)	397 (0.34s)	1003 (0.27s)
<i>pseudoobscura</i>	0.0		743 (3.13%)	743 (2.14%)	724 (1.40%)	912 (3.76%)
	0.3		836 (1.06s)	849 (1.03s)	837 (0.96s)	980 (1.06s)
	0.5		929 (0.44s)	938 (0.43s)	908 (0.43s)	1015 (0.39s)
<i>sechellia</i>	0.0			171 (45.95s)	236 (6.41s)	850 (2.19%)
	0.3			194 (1.06s)	301 (0.74s)	982 (2.11s)
	0.5			244 (0.44s)	423 (0.40s)	1014 (0.28s)
<i>simulans</i>	0.0				265 (18.14s)	863 (2.35%)
	0.3				336 (0.63s)	994 (0.87s)
	0.5				453 (0.33s)	1005 (0.25s)
<i>yakuba</i>	0.0					830 (1.73%)
	0.3					972 (0.72s)
	0.5					992 (0.24s)

■ **Table 7** Computed $\text{ffd}_{\text{DCJ}}^{\text{DP}}$ and elapsed time (or gap in %) in pairwise comparisons of X chromosomes, with cutting thresholds ranging between $x = 0.0$ and $x = 0.3$. The time limit is 3600s.

species	x	<i>pseudoobscura</i>	<i>sechellia</i>	<i>simulans</i>	<i>yakuba</i>	<i>busckii</i>
<i>melanogaster</i>	0.0	1390.3 (255.22s)	407.3 (0.32%)	432.4 (9.60s)	587.4 (4.21s)	1362.7 (109.59s)
	0.1	1370.1 (30.01s)	408.3 (0.34%)	433.8 (10.12s)	590.0 (4.08s)	1363.9 (11.49s)
	0.2	1326.0 (6.28s)	412.7 (174.06s)	436.6 (5.25s)	601.0 (2.17s)	1344.7 (5.12s)
	0.3	1296.0 (4.13s)	416.7 (24.47s)	445.4 (3.55s)	609.3 (1.64s)	1321.8 (2.87s)
<i>pseudoobscura</i>	0.0		1417.1 (258.89s)	1375.6 (281.95s)	1361.7 (94.68s)	1515.7 (368.78s)
	0.1		1394.1 (36.51s)	1355.1 (45.6s)	1337.0 (17.74s)	1491.7 (33.27s)
	0.2		1344.1 (3.64s)	1309.7 (329.25s)	1299.5 (3.34s)	1433.3 (5.61s)
	0.3		1308.0 (5.56s)	1278.0 (3.73s)	1262.3 (3.69s)	1374.1 (4.69s)
<i>sechellia</i>	0.0			352.5 (5.90s)	626.8 (4.70s)	1378.2 (74.01s)
	0.1			352.8 (5.83s)	630.4 (3.92s)	1377.1 (23.36s)
	0.2			351.9 (3.56s)	635.3 (3.08s)	1354.2 (5.38s)
	0.3			355.0 (2.55s)	641.3 (1.92s)	1328.3 (4.18s)
<i>simulans</i>	0.0				617.8 (7.78s)	1344.0 (80.84s)
	0.1				621.3 (5.27s)	1342.7 (29.58s)
	0.2				626.2 (2.04s)	1316.7 (5.50s)
	0.3				637.8 (1.99s)	1295.5 (3.25s)
<i>yakuba</i>	0.0					1325.5 (69.40s)
	0.1					1323.7 (24.32s)
	0.2					1304.9 (6.27s)
	0.3					1280.8 (3.73s)

■ **Table 8** Association between genes in pairwise comparisons of the corresponding X chromosomes, considering pairwise similarities strictly greater than 0. For the three tables, the element stored in line i and column j represents the number of genes of the species i in the pairwise comparison of genomes i and j . The X chromosome of *D. pseudoobscura* was fused with another chromosome during evolution [17], therefore it presents a larger number of unassociated genes when compared to the other species.

Number of unassociated genes

species		<i>melanog</i>	<i>pseudoob</i>	<i>sechellia</i>	<i>simulans</i>	<i>yakuba</i>	<i>busckii</i>
	#genes	2043	4770	2107	2007	1956	1953
<i>melanogaster</i>	2043	—	152	23	84	100	221
<i>pseudoobscura</i>	4770	2076	—	2025	2127	2113	2110
<i>sechellia</i>	2107	57	174	—	102	133	257
<i>simulans</i>	2007	84	187	74	—	130	272
<i>yakuba</i>	1956	80	153	80	107	—	227
<i>busckii</i>	1953	167	124	173	217	201	—

Number of genes uniquely associated

species		<i>melanog</i>	<i>pseudoob</i>	<i>sechellia</i>	<i>simulans</i>	<i>yakuba</i>	<i>busckii</i>
	#genes	2043	4770	2107	2007	1956	1953
<i>melanogaster</i>	2043	—	1052	1440	1402	1428	1191
<i>pseudoobscura</i>	4770	1613	—	1651	1579	1646	1565
<i>sechellia</i>	2107	1479	1084	—	1454	1449	1224
<i>simulans</i>	2007	1382	1024	1392	—	1352	1126
<i>yakuba</i>	1956	1352	1017	1353	1331	—	1123
<i>busckii</i>	1953	1155	977	1154	1125	1147	—

Number of genes associated to at least two other genes

species		<i>melanog</i>	<i>pseudoob</i>	<i>sechellia</i>	<i>simulans</i>	<i>yakuba</i>	<i>busckii</i>
	#genes	2043	4770	2107	2007	1956	1953
<i>melanogaster</i>	2043	—	839	580	557	515	631
<i>pseudoobscura</i>	4770	1081	—	1094	1064	1011	1095
<i>sechellia</i>	2107	571	849	—	551	525	626
<i>simulans</i>	2007	541	796	541	—	525	609
<i>yakuba</i>	1956	524	786	523	518	—	606
<i>busckii</i>	1953	631	852	626	611	605	—