CHAPTER 3

DCJ model of canonical genomes

Let $\mathbb{A}$ and $\mathbb{B}$ be two annotated genomes and note that, if $\mathbb{A}$ and $\mathbb{B}$ form a canonical pair, then $\mathcal{F}(\mathbb{A}) = \mathcal{G}(\mathbb{A}) = \mathcal{G}(\mathbb{B}) = \mathcal{F}(\mathbb{B})$. We then denote by n_* the cardinality of all these sets: $n_* = |\mathcal{F}(\mathbb{A})| = |\mathcal{G}(\mathbb{A})| = |\mathcal{G}(\mathbb{B})| = |\mathcal{F}(\mathbb{B})|$. Recall that, in this case, only DCJ operations are used for sorting one genome into the other, and the corresponding DCJ distance is denoted by $\mathrm{d}_{\mathrm{DCJ}}(\mathbb{A}, \mathbb{B})$.

3.1 Relational graph of canonical genomes

Finding sorting DCJ operations and computing the DCJ distance between two canonical genomes $\mathbb{A}$ and $\mathbb{B}$ can be achieved with the help of the *relational graph* of $\mathbb{A}$ and $\mathbb{B}$ [18], denoted by $\mathsf{G}_{\mathrm{R}}(\mathbb{A}, \mathbb{B}) = (V, E)$, whose sets of vertices and edges are defined as follows:

1. The set of vertices is $V = V(\mathbb{A}) \cup V(\mathbb{B})$, where

   $V(\mathbb{A})$ contains a vertex for each extremity of each marker in $\mathcal{M}(\mathbb{A})$ and
   $V(\mathbb{B})$ contains a vertex for each extremity of each marker in $\mathcal{M}(\mathbb{B})$.

   Each vertex v has an identifier corresponding to the unannotated marker extremity it represents, and a label $\eta(v)$, corresponding to the annotated marker extremity it represents. Note that there are $4n_*$ vertices in $\mathsf{G}_{\mathrm{R}}(\mathbb{A}, \mathbb{B})$, $2n_*$ per genome.

2. The set of edges is $E = E_{\mathsf{adj}}(\mathbb{A}) \cup E_{\mathsf{adj}}(\mathbb{B}) \cup E_{\mathsf{ext}}$, where the *adjacency edges* are sets

   $$\begin{aligned} E_{\mathsf{adj}}(\mathbb{A}) = &\ \{uv : u, v \in V(\mathbb{A}) \text{ and } uv \in \mathsf{adj}(\mathbb{A})\} \text{ and} \\ E_{\mathsf{adj}}(\mathbb{B}) = &\ \{uv : u, v \in V(\mathbb{B}) \text{ and } uv \in \mathsf{adj}(\mathbb{B})\}, \end{aligned}$$

and the set of *extremity edges* (whose cardinality is $2n_*$) is

$$E_{\mathsf{ext}} = \{uv : u \in V(\mathbb{A}) \text{ and } v \in V(\mathbb{B}) \text{ and } \eta(u) = \eta(v)\}.$$

Since any vertex in $\mathsf{G}_{\mathrm{R}}(\mathbb{A}, \mathbb{B})$ has exactly one extremity edge and at most one adjacency edge, its degree is one or two. Therefore, $\mathsf{G}_{\mathrm{R}}(\mathbb{A}, \mathbb{B})$ is a collection of paths and cycles. A vertex (marker extremity) that has no adjacency edge corresponds to a telomere and is therefore also called *telomere*. Each cutpoint of each genome is represented in the graph either as an adjacency edge or as a telomere. Recall that the number of cutpoints in $\mathbb{A}$ (respectively $\mathbb{B}$) is $n_* + \kappa(\mathbb{A})$ (respectively $n_* + \kappa(\mathbb{B})$).

Each connected component of the graph alternates between extremity edges and cutpoints, and we define the *length* of a component Γ to be the number of extremity edges in Γ. An *i-cycle* and an *i-path* denote respectively a cycle and a path of length i. Note that all cycles have even length, while paths start and end with extremity edges and can have even or odd lengths, called *even* and *odd paths* respectively.

A cycle can be simply denoted by $\mathbb{O}$. An odd path has one endpoint in a telomere from $\mathbb{A}$ and the other endpoint in a telomere from $\mathbb{B}$ and is called an $\mathbb{AB}$-path, simply denoted by $\mathbb{AB}$. Even paths have either both endpoints in $\mathbb{A}$, being an $\mathbb{AA}$-path, simply denoted by $\mathbb{AA}$, or both endpoints in $\mathbb{B}$, being a $\mathbb{BB}$-path, simply denoted by $\mathbb{BB}$. Even paths can also be called *unbalanced paths*, while odd paths are also called *balanced paths*. Let the sets of cycles, $\mathbb{AB}$-, $\mathbb{AA}$- and $\mathbb{BB}$-paths be respectively denoted by $\mathcal{C}$, $\mathcal{P}_{\mathbb{AB}}$, $\mathcal{P}_{\mathbb{AA}}$ and $\mathcal{P}_{\mathbb{BB}}$. Now let $\Upsilon(\Gamma)$ give the type of component Γ. For example, if Γ is a cycle, then $\Upsilon(\Gamma) = \mathbb{O}$. We can then explicitly write the above mentioned sets as:

$$\mathcal{C} = \{\Gamma \mid \Gamma \in \mathsf{G}_{\mathrm{R}}(\mathbb{A}, \mathbb{B}) \text{ and } \Upsilon(\Gamma) = \mathbb{O}\} ,$$

$$\mathcal{P}_{\mathbb{AB}} = \{\Gamma \mid \Gamma \in \mathsf{G}_{\mathrm{R}}(\mathbb{A}, \mathbb{B}) \text{ and } \Upsilon(\Gamma) = \mathbb{AB}\} ,$$

$$\mathcal{P}_{\mathbb{AA}} = \{\Gamma \mid \Gamma \in \mathsf{G}_{\mathrm{R}}(\mathbb{A}, \mathbb{B}) \text{ and } \Upsilon(\Gamma) = \mathbb{AA}\} \text{ and}$$

$$\mathcal{P}_{\mathbb{BB}} = \{\Gamma \mid \Gamma \in \mathsf{G}_{\mathrm{R}}(\mathbb{A}, \mathbb{B}) \text{ and } \Upsilon(\Gamma) = \mathbb{BB}\}.$$

Recall that $\kappa(\mathbb{A})$ and $\kappa(\mathbb{B})$ are the numbers of linear chromosomes in genomes $\mathbb{A}$ and $\mathbb{B}$. The endpoints of paths and chromosomes are the same telomeres, therefore we have $\kappa(\mathbb{A})+\kappa(\mathbb{B}) = |\mathcal{P}_{\mathbb{AB}}|+|\mathcal{P}_{\mathbb{AA}}|+|\mathcal{P}_{\mathbb{BB}}|$. Furthermore, the numbers of telomeres in each genome are even. Since each $\mathbb{AA}$- or $\mathbb{BB}$-path takes either zero or two telomeres per genome and each $\mathbb{AB}$-path takes one telomere per genome, the number of $\mathbb{AB}$-paths must be even.

Related graphs. As illustrated in Figure 3.1, the relational graph has the same properties of two simpler graphs that were proposed earlier:

1. The first is the so-called *breakpoint graph*, originally proposed in the seminal studies of the *inversion sorting and distance* [7]. It can be derived from the relational graph by contracting each extremity edge e of $\mathsf{G}_{\mathrm{R}}(\mathbb{A}, \mathbb{B}) = (V, E)$ and assigning to the resulting single vertex the common annotation of the vertices that are connected by e. In the breakpoint graph there are only adjacency edges. Furthermore, cycles also have even length, while $\mathbb{AB}$-paths are even and $\mathbb{AA}$- and $\mathbb{BB}$-paths are odd.

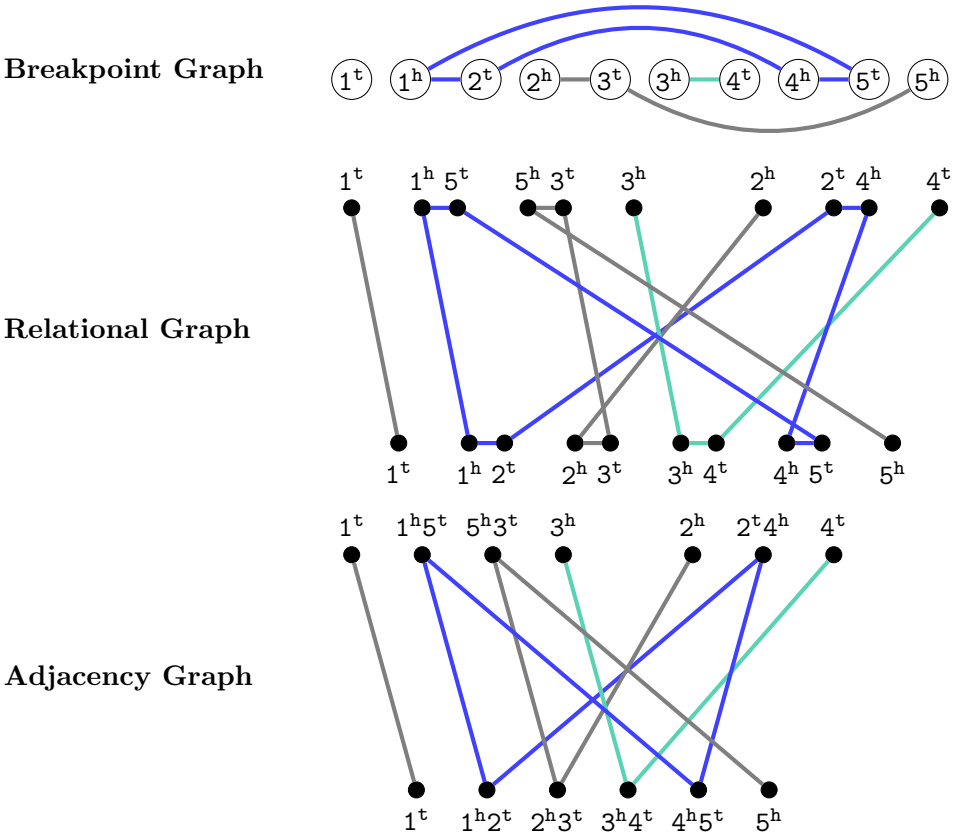

Figure 3.1: For a canonical pair formed by multilinear genome $\mathbb{A} = \{ [1\ 5\ 3], [\overline{2}\ \overline{4}] \}$ and unilinear genome $\mathbb{B} = \{ [1\ 2\ 3\ 4\ 5] \}$, where $n_* = 5$, we represent the relational graph (in the middle) surrounded by the breakpoint graph (top) and by the adjacency graph (bottom). Note that the number of vertices in the breakpoint graph and the numbers of edges in both relational and adjacency graphs are equal to $2n_*$. In all graphs we have a (blue) 4-cycle, a (red) $\mathbb{AA}$-path and two $\mathbb{AB}$-paths.

2. The second is the so-called *adjacency graph*, which is bipartite and was originally proposed in the formalization of the *DCJ sorting and distance* [10]. It can be derived from the relational graph by contracting each adjacency edge a of $\mathsf{G}_\mathrm{R}(\mathbb{A}, \mathbb{B}) = (V, E)$, concatenating in the label of the resulting single vertex the annotations of the vertices that are connected by a. In other words, the vertices of the adjacency graph are the adjacencies and telomeres of $\mathbb{A}$ and $\mathbb{B}$ and all edges are extremity edges. Similarly to the relational graph, in the adjacency graph cycles have even length, $\mathbb{AB}$-paths are odd and $\mathbb{AA}$- and $\mathbb{BB}$-paths are even.

Relational graph of sorted and unsorted genomes. The smallest components that can occur in $\mathsf{G}_\mathrm{R}(\mathbb{A}, \mathbb{B})$ are 2-cycles and ($\mathbb{AB}$) 1-paths, denoted *short components*. A cycle whose length is greater than 2 or a path whose length is greater than 1 is called a *long component*. When canonical genomes $\mathbb{A}$ and $\mathbb{B}$ are identical (or *sorted*), their relational graph is a collection of short components: identical genomes have the same sets of adjacencies and telomeres,

and each common adjacency corresponds to a 2-cycle while each common telomere corresponds to a 1-path in $\mathsf{G}_{\mathrm{R}}(\mathbb{A}, \mathbb{B})$. Recall that the length of a component corresponds to its number of extremity edges and that $\mathsf{G}_{\mathrm{R}}(\mathbb{A}, \mathbb{B})$ has $2n_*$ extremity edges. Therefore, for sorted genomes we have $2n_* = 2|\mathcal{C}| + |\mathcal{P}_{\mathbb{AB}}|$ and, consequently, $n_* = |\mathcal{C}| + \frac{|\mathcal{P}_{\mathbb{AB}}|}{2}$. Otherwise, when canonical genomes $\mathbb{A}$ and $\mathbb{B}$ are distinct (or *unsorted*), their relational graph contains at least one long component. Therefore, in this case $n_* > |\mathcal{C}| + \frac{|\mathcal{P}_{\mathbb{AB}}|}{2}$. With these observations we can already see that the DCJ operations that transform $\mathbb{A}$ into $\mathbb{B}$ must increase the numbers of cycles and/or of $\mathbb{AB}$-paths in $\mathsf{G}_{\mathrm{R}}(\mathbb{A}, \mathbb{B})$. In the following we will present the results from Bergeron *et al.* [10], explaining how this can be achieved.

3.2 Types of DCJ operation with respect to the relational graph

Note that, with respect to its effect on the relational graph, a DCJ ρ cuts one or two components, and rejoins the open ends to transform them into one or two new components. A DCJ operation ρ is said to be *internal* (INT) to a single component Γ, when ρ cuts only at cutpoint(s) that are in Γ. The result of an internal DCJ ρ can be one component (distinct from Γ) or two components. In contrast, A DCJ operation ρ is said to be a *recombination* (REC) when ρ cuts at cutpoints of two distinct components Γ and Γ'. The result of a recombination ρ can be either a single component or two components (distinct from Γ and Γ').

In order to describe all possible DCJ operations, we adopt the following notation to represent the types of components and their lengths:

- $\mathbb{O}^{\langle\imath\rangle}$: even cycle with length $\imath \in \{2, 4, 6, \ldots\}$;
- $\mathbb{AB}^{\langle\imath\rangle}$: balanced path with length $\imath \in \{1, 3, 5, \ldots\}$;
- $\mathbb{AA}^{\langle\imath\rangle}$: unbalanced $\mathbb{AA}$-path with length $\imath \in \{2, 4, \ldots\}$;
- $\mathbb{BB}^{\langle\imath\rangle}$: unbalanced $\mathbb{BB}$-path with length $\imath \in \{2, 4, \ldots\}$;
- $\Gamma^{\langle\imath\rangle}$: component of any type with length $\imath \geq 1$.

The possible types of DCJ operation applied on cutpoints of genome $\mathbb{A}$ are described in Table 3.1 [10]. Note that each DCJ operation must be of one of three types:

Gaining DCJ. Either increases $|\mathcal{C}|$ by one or increases $|\mathcal{P}_{\mathbb{AB}}|$ by two.

Neutral DCJ. Does not change the cardinalities of the sets $\mathcal{C}$ and $\mathcal{P}_{\mathbb{AB}}$.

Losing DCJ. Either decreases $|\mathcal{C}|$ by one or decreases $|\mathcal{P}_{\mathbb{AB}}|$ by two.

Table 3.1: Effects of DCJ operations (applied on cutpoints of $\mathbb{A}$) on the relational graph

	component(s) [state]	DCJ	component(s) [state]		affects...
1.	$\boxed{\Gamma^{\langle \iota+\jmath \rangle}}$ $[\textcircled{r}^4 / \textcircled{r}^3]$	gaining (INT) ⇌ losing (REC)	$\boxed{\mathbb{O}^{\langle \iota \rangle}\ \Gamma^{\langle \jmath \rangle}}$ $[\textcircled{r}^4 / \textcircled{r}^3]$	with $\begin{cases} \iota \in \{2,4,6,\ldots\} \\ \jmath \geq 1 \\ \Upsilon(\Gamma^{\langle \iota+\jmath \rangle}) = \Upsilon(\Gamma^{\langle \jmath \rangle}) \end{cases}$	$\Delta\lvert\mathcal{C}\rvert$ by ± 1
2.	$\boxed{\Gamma^{\langle \iota+\jmath \rangle}}$ $[\textcircled{r}^4 / \textcircled{r}^3]$	neutral (INT) ⇌ neutral (INT)	$\boxed{\check{\Gamma}^{\langle \iota+\jmath \rangle}}$ $[\textcircled{r}^4 / \textcircled{r}^3]$	with $\begin{cases} \iota \in \{2,4,6,\ldots\} \\ \jmath \geq 1 \\ \Gamma^{\langle \iota+\jmath \rangle}) \neq \check{\Gamma}^{\langle \iota+\jmath \rangle} \\ \Upsilon(\Gamma^{\langle \iota+\jmath \rangle}) = \Upsilon(\check{\Gamma}^{\langle \iota+\jmath \rangle}) \end{cases}$	
3.	$\boxed{\mathbb{AA}^{\langle \iota+\jmath \rangle}\ \mathbb{BB}^{\langle \iota'+\jmath' \rangle}}$ $[\textcircled{r}^4 / \textcircled{r}^3]$	gaining (INT) ⇌ losing (REC)	$\boxed{\mathbb{AB}^{\langle \iota+\iota' \rangle}\ \mathbb{AB}^{\langle \jmath+\jmath' \rangle}}$ $[\textcircled{r}^4 / \textcircled{r}^3]$	with $\begin{cases} \iota \in \{0,2,4,\ldots\} \\ \jmath \in \{2,4,6,\ldots\} \\ \iota', \jmath' \in \{1,3,5,\ldots\} \end{cases}$	$\Delta\lvert\mathcal{P}_{\mathbb{AB}}\rvert$ by ± 2
4.	$\boxed{\mathbb{AB}^{\langle \iota+\jmath' \rangle}\ \mathbb{AB}^{\langle \jmath+\iota' \rangle}}$ $[\textcircled{r}^4 / \textcircled{r}^3]$	neutral (REC) ⇌ neutral (REC)	$\boxed{\mathbb{AB}^{\langle \iota+\iota' \rangle}\ \mathbb{AB}^{\langle \jmath+\jmath' \rangle}}$ $[\textcircled{r}^4 / \textcircled{r}^3]$	with $\begin{cases} \text{all } \mathbb{AB}\text{-paths distinct} \\ \iota \in \{0,2,4,\ldots\} \\ \jmath \in \{2,4,6,\ldots\} \\ \iota', \jmath' \in \{1,3,5,\ldots\} \end{cases}$	
5.	$\boxed{\mathbb{AA}^{\langle \iota+\jmath \rangle}\ \mathbb{AB}^{\langle \iota'+\jmath' \rangle}}$ $[\textcircled{r}^4 / \textcircled{r}^3]$	neutral (REC) ⇌ neutral (REC)	$\boxed{\mathbb{AA}^{\langle \iota+\iota' \rangle}\ \mathbb{AB}^{\langle \jmath+\jmath' \rangle}}$ $[\textcircled{r}^4 / \textcircled{r}^3]$	with $\begin{cases} \text{all paths distinct} \\ \iota, \iota' \in \{2,4,6,\ldots\} \\ \jmath \in \{0,2,4,\ldots\} \\ \jmath' \in \{1,3,5,\ldots\} \end{cases}$	
6.	$\boxed{\mathbb{BB}^{\langle \iota+\jmath \rangle}\ \mathbb{AB}^{\langle \iota'+\jmath' \rangle}}$ $[\textcircled{r}^4 / \textcircled{r}^3]$	neutral (REC) ⇌ neutral (REC)	$\boxed{\mathbb{BB}^{\langle \iota+\jmath' \rangle}\ \mathbb{AB}^{\langle \iota'+\jmath \rangle}}$ $[\textcircled{r}^4 / \textcircled{r}^3]$	with $\begin{cases} \text{all paths distinct} \\ \iota, \jmath, \jmath' \in \{1,3,5,\ldots\} \\ \iota' \in \{0,2,4,\ldots\} \end{cases}$	
7.	$\boxed{\mathbb{AA}^{\langle \iota+\jmath' \rangle}\ \mathbb{AA}^{\langle \jmath+\iota' \rangle}}$ $[\textcircled{r}^4 / \textcircled{r}^3]$	neutral (REC) ⇌ neutral (REC)	$\boxed{\mathbb{AA}^{\langle \iota+\iota' \rangle}\ \mathbb{AA}^{\langle \jmath+\jmath' \rangle}}$ $[\textcircled{r}^4 / \textcircled{r}^3]$	with $\begin{cases} \text{all } \mathbb{AA}\text{-paths distinct} \\ \iota \in \{0,2,4,\ldots\} \\ \jmath, \iota', \jmath' \in \{2,4,6,\ldots\} \end{cases}$	
8.	$\boxed{\mathbb{BB}^{\langle \iota+\jmath' \rangle}\ \mathbb{BB}^{\langle \jmath+\iota' \rangle}}$ $[\textcircled{r}^4]$	neutral (REC) ⇌ neutral (REC)	$\boxed{\mathbb{BB}^{\langle \iota+\iota' \rangle}\ \mathbb{BB}^{\langle \jmath+\jmath' \rangle}}$ $[\textcircled{r}^4]$	with $\begin{cases} \text{all } \mathbb{BB}\text{-paths distinct} \\ \iota, \jmath, \iota', \jmath' \in \{1,3,5,\ldots\} \end{cases}$	
9.	$\boxed{\mathbb{AA}^{\langle \iota \rangle}}$ $[\textcircled{r}^{2\lvert 2}]$	gaining (INT) ⇌ losing (INT)	$\boxed{\mathbb{O}^{\langle \iota \rangle}}$ $[\textcircled{r}^{2\lvert 1}]$	with $\iota \in \{2,4,6,\ldots\}$	$\Delta\lvert\mathcal{C}\rvert$ by ± 1
10.	$\boxed{\mathbb{BB}^{\langle \iota+\jmath \rangle}}$ $[\textcircled{r}^{2\lvert 1}]$	gaining (REC) ⇌ losing (REC)	$\boxed{\mathbb{AB}^{\langle \iota \rangle}\ \mathbb{AB}^{\langle \jmath \rangle}}$ $[\textcircled{r}^{2\lvert 2}]$	with $\iota, \jmath \in \{1,3,5,\ldots\}$	$\Delta\lvert\mathcal{P}_{\mathbb{AB}}\rvert$ by ± 2
11.	$\boxed{\mathbb{AA}^{\langle \iota+\jmath \rangle}}$ $[\textcircled{r}^{2\lvert 1}]$	neutral (INT) ⇌ neutral (REC)	$\boxed{\mathbb{AA}^{\langle \iota \rangle}\ \mathbb{AA}^{\langle \jmath \rangle}}$ $[\textcircled{r}^{2\lvert 2}]$	with $\iota, \jmath \in \{2,4,6,\ldots\}$	
12.	$\boxed{\mathbb{AB}^{\langle \iota+\jmath \rangle}}$ $[\textcircled{r}^{2\lvert 1}]$	neutral (INT) ⇌ neutral (REC)	$\boxed{\mathbb{AA}^{\langle \iota \rangle}\ \mathbb{AB}^{\langle \jmath \rangle}}$ $[\textcircled{r}^{2\lvert 2}]$	with $\begin{cases} \iota \in \{2,4,6\ldots\} \\ \jmath \in \{1,3,5,\ldots\} \end{cases}$	

3.3 DCJ distance formula

Gaining DCJ operations lead to a "sorted" graph. In a sorting procedure it is always possible to find a gaining DCJ at each step, therefore any optimal operation is gaining [10]. This can be verified simply by looking at Table 3.1: it is obvious that an unsorted graph will have at least one of the situations leading to the gaining operations shown in lines 1, 3, 9 and 10.

Since gaining operations are those that produce the maximum possible increase of cycles or $\mathbb{AB}$-paths, the following theorem holds.

Theorem 1 ([10]) *Given a canonical pair of genomes $\mathbb{A}$ and $\mathbb{B}$, their DCJ distance is*

$$\mathrm{d}_{\mathrm{DCJ}}(\mathbb{A},\mathbb{B}) = n_* - \left(|\mathcal{C}| + \frac{|\mathcal{P}_{\mathbb{AB}}|}{2}\right).$$

3.4 Triangular inequality

Given any three canonical genomes $\mathbb{A}$, $\mathbb{B}$ and $\mathbb{C}$, consider without loss of generality that $\mathrm{d}_{\mathrm{DCJ}}(\mathbb{A},\mathbb{B}) \geq \mathrm{d}_{\mathrm{DCJ}}(\mathbb{A},\mathbb{C})$ and $\mathrm{d}_{\mathrm{DCJ}}(\mathbb{A},\mathbb{B}) \geq \mathrm{d}_{\mathrm{DCJ}}(\mathbb{B},\mathbb{C})$. Then the triangular inequality is the property that guarantees that $\mathrm{d}_{\mathrm{DCJ}}(\mathbb{A},\mathbb{B}) \leq \mathrm{d}_{\mathrm{DCJ}}(\mathbb{A},\mathbb{C}) + \mathrm{d}_{\mathrm{DCJ}}(\mathbb{B},\mathbb{C})$. It obviously holds for the DCJ distance: by combining a sorting scenario from $\mathbb{A}$ to $\mathbb{C}$ with $\mathrm{d}_{\mathrm{DCJ}}(\mathbb{A},\mathbb{C})$ steps and a sorting scenario from $\mathbb{C}$ to $\mathbb{B}$ with $\mathrm{d}_{\mathrm{DCJ}}(\mathbb{B},\mathbb{C})$ steps we trivially get a sorting scenario from $\mathbb{A}$ to $\mathbb{B}$ with $\mathrm{d}_{\mathrm{DCJ}}(\mathbb{A},\mathbb{C}) + \mathrm{d}_{\mathrm{DCJ}}(\mathbb{B},\mathbb{C})$ steps. Therefore, it is clear that $\mathrm{d}_{\mathrm{DCJ}}(\mathbb{A},\mathbb{B})$ cannot be greater than $\mathrm{d}_{\mathrm{DCJ}}(\mathbb{A},\mathbb{C}) + \mathrm{d}_{\mathrm{DCJ}}(\mathbb{B},\mathbb{C})$, otherwise it would contradict the fact that it corresponds to the length of a most parsimonious sorting scenario.

3.5 On DCJ sorting

Proposition 1 *For any DCJ-state of type $\textcircled{r}^4$ or $\textcircled{r}^3$ or $\textcircled{r}^{2|2}$ whose cutpoints are in genome $\mathbb{A}$ and belong to the same long component Γ of $\mathsf{G}_{\mathrm{R}}(\mathbb{A},\mathbb{B})$, there is one, and only one (internal) gaining DCJ operation.*

Proof: For each pair of $\mathbb{A}$-cutpoints, such that at most one is a telomere, we know that there are two different DCJ operations. When the cutpoints belong to the same component Γ, one of the two operations simply inverts a fragment, not changing the structure of Γ, and is therefore neutral (Table 3.1, line 2.). The second operation is a gaining DCJ that splits Γ into a cycle and a smaller component of the same type as Γ, increasing the number of cycles (Table 3.1, line 1.). This includes all pairs of $\mathbb{A}$-cutpoints in cycles, $\mathbb{AB}$-paths and $\mathbb{BB}$-paths, and all pairs of $\mathbb{A}$-cutpoints in $\mathbb{AA}$-paths excluding the case where the two cutpoints are telomeres. For the particular case where the two cutpoints are telomeres of an $\mathbb{AA}$-path forming a DCJ-state of type $\textcircled{r}^{2|2}$, there is only one (internal) DCJ operation, and this operation is gaining (Table 3.1, line 9). □

Proposition 1 guarantees that there is a gaining DCJ operation for each DCJ-state of type $\textcircled{r}^4$ or $\textcircled{r}^3$ or $\textcircled{r}^{2|2}$ internal to any long component of $\mathsf{G}_{\mathrm{R}}(\mathbb{A},\mathbb{B})$. In particular, this DCJ-state can be a pair of cutpoints directly connected to an adjacency α in genome $\mathbb{B}$. If the result

of this particular cut is to join such that the adjacency α is created in $\mathbb{A}$, a new 2-cycle appears in the relational graph, meaning that a DCJ operation that creates an adjacency of genome $\mathbb{B}$ in genome $\mathbb{A}$ is gaining.

Corollary 1 *Let α be an adjacency of $\mathbb{B}$ that is not present in $\mathbb{A}$. The DCJ operation that reconstructs α in $\mathbb{A}$ is gaining.*

Once all adjacencies of $\mathbb{B}$ are reconstructed in $\mathbb{A}$, the only long components that can exist in the relational graph are $\mathbb{BB}$-paths of length 2. Any $\mathbb{BB}$-path of length 2 can be split into two $\mathbb{AB}$-paths with a gaining DCJ (Table 3.1 line 10):

Corollary 2 *For any DCJ-state of type $\textcircled{r}^{2|1}$ whose cutpoint is an $\mathbb{A}$-adjacency of a $\mathbb{BB}$-path there is only one (internal) DCJ operation, and this operation is gaining.*

The results above give a simple greedy algorithm to find one optimal sequence of DCJ operations (exclusively composed of gaining DCJs) to sort $\mathbb{A}$ into $\mathbb{B}$ consisting of simply reconstructing each adjacency of $\mathbb{B}$ that is not in $\mathbb{A}$, followed by reconstructing each telomere of $\mathbb{B}$ that is not in $\mathbb{A}$ [10].

3.6 Complexity of DCJ distance and greedy sorting

The cutpoints of genome $\mathbb{A}$ with n_* markers can be obtained by reading the chromosomes of $\mathbb{A}$ once and storing the cutpoints in a vector of length $n_* + \kappa(\mathbb{A})$. Simultaneously, we can store in another vector of length $2n_*$ a pointer to the cutpoint to which each marker extremity belongs. The cutpoints and pointers of genome $\mathbb{B}$ can be obtained similarly. Since $\kappa(\mathbb{A}) \leq n_*$ and $\kappa(\mathbb{B}) \leq n_*$, this procedure takes $O(n_*)$ time and space.

These four vectors are an implict representation of the relational graph: by navigating on them we can easily obtain the cycles and paths of the graph, still in $O(n_*)$.

Finally, for greedily sorting we only need to visit the adjacencies of genome $\mathbb{B}$ followed by the telomeres of genome $\mathbb{B}$ and reconstruct one by one in genome $\mathbb{A}$. Since each of these reconstructions requires constant time access to the vetcors of $\mathbb{A}$, the greedy sorting can be easily done in $O(n_*)$ time and space [10].

3.7 Capping the canonical relational graph optimally

The paths of the relational graph can be converted into cycles with a technique called *capping* [55]. It consists of modifying the graph by adding *artificial* extremities, called *cap extremities*, that link all paths into cycles. Each telomere must be connected to a distinct cap extremity by an additional *semi-artificial adjacency edge*. Furthermore, two cap extremities that are in the same genome and not connected to telomeres can be connected to each other by an *artificial adjacency edge*. Then, each cap extremity in genome $\mathbb{A}$ must be connected to a distinct cap extremity in genome $\mathbb{B}$ by a cap extremity edge. Therefore, the capping always connect a telomere in genome $\mathbb{A}$ to a telomere in genome $\mathbb{B}$ via two cap extremities.

Since the number of telomeres in each genome is even, the number of cap extremities is also even. Each pair of cap extremities compose an "artificial" marker called *cap*.

There many ways of performing a capping. A procedure for obtaining an *optimal capping*, that preserves the DCJ distance, is given in the following.

Optimal capping of $\mathbb{AB}$-paths. For each $\mathbb{AB}$-path whose telomeres are $\gamma_{\mathbb{A}}$ and $\gamma_{\mathbb{B}}$, add cap extremity vertices $\circ_{\mathbb{A}}$ and $\circ_{\mathbb{B}}$ and connect with (semi-artificial) adjacency edges $\circ_{\mathbb{A}}$ to $\gamma_{\mathbb{A}}$ and $\circ_{\mathbb{B}}$ to $\gamma_{\mathbb{B}}$. Furthermore, connect with a (cap) extremity edge the vertices $\circ_{\mathbb{A}}$ to $\circ_{\mathbb{B}}$. Note that this removes one $\mathbb{AB}$-path, adds one cycle, but also adds half a cap to the set of markers. Since the number of $\mathbb{AB}$-paths is even, each pair of capped $\mathbb{AB}$-paths removes two $\mathbb{AB}$-paths, adds two cycles, but also adds one cap to the set of markers. We denote the capping of path $\mathbb{AB}$ by $\zeta(\mathbb{AB})$.

Optimal capping of $\mathbb{AA}$- and $\mathbb{BB}$-paths. Let an $\mathbb{AA}$-path have telomeres $\gamma^1_{\mathbb{A}}$ and $\gamma^2_{\mathbb{A}}$ and a $\mathbb{BB}$-path have telomeres $\gamma^1_{\mathbb{B}}$ and $\gamma^2_{\mathbb{B}}$. These paths can be optimally linked together into a single cycle as follows. Add cap extremity vertices $\circ^1_{\mathbb{A}}$, $\circ^2_{\mathbb{A}}$, $\circ^1_{\mathbb{B}}$ and $\circ^2_{\mathbb{B}}$ and connect with (semi-artificial) adjacency edges $\circ^1_{\mathbb{A}}$ to $\gamma^1_{\mathbb{A}}$, $\circ^2_{\mathbb{A}}$ to $\gamma^2_{\mathbb{A}}$, $\circ^1_{\mathbb{B}}$ to $\gamma^1_{\mathbb{B}}$, and $\circ^2_{\mathbb{B}}$ to $\gamma^2_{\mathbb{B}}$. Furthermore, connect with (cap) extremity edges $\circ^1_{\mathbb{A}}$ to $\circ^1_{\mathbb{B}}$ and $\circ^2_{\mathbb{A}}$ to $\circ^2_{\mathbb{B}}$. Note that this adds one cycle, but also adds one *cap* to the set of markers. We denote the capping of paths $\mathbb{AA}$ and $\mathbb{BB}$ together by $\zeta(\mathbb{AA}, \mathbb{BB})$. Alternatively, each $\mathbb{AA}$- or $\mathbb{BB}$-path can also be separately capped into a cycle, and this alternative is also optimal. Again, let an $\mathbb{AA}$-path have telomeres $\gamma^1_{\mathbb{A}}$ and $\gamma^2_{\mathbb{A}}$. Add cap extremity vertices $\circ^1_{\mathbb{A}}$, $\circ^2_{\mathbb{A}}$, $\circ^2_{\mathbb{A}}$ and $\circ^2_{\mathbb{B}}$ and connect with (semi-artificial) adjacency edges $\circ^1_{\mathbb{A}}$ to $\gamma^1_{\mathbb{A}}$ and $\circ^2_{\mathbb{A}}$ to $\gamma^2_{\mathbb{A}}$. Furthermore, connect with an (artificial) adjacency edge the two cap vertices $\circ^1_{\mathbb{B}}$ and $\circ^2_{\mathbb{B}}$ and with cap extremity edges $\circ^1_{\mathbb{A}}$ to $\circ^1_{\mathbb{B}}$ and $\circ^2_{\mathbb{A}}$ to $\circ^2_{\mathbb{B}}$. Note that this adds one cycle, but also adds one *cap* to the set of markers. Capping a $\mathbb{BB}$-path separately can be done analogously. We denote the cappings of path $\mathbb{AA}$ (respectively $\mathbb{BB}$) by $\zeta(\mathbb{AA}, \mathring{\alpha}_{\mathbb{B}})$ (respectively by $\zeta(\mathbb{BB}, \mathring{\alpha}_{\mathbb{A}})$).

Table 3.2 summarizes the effect of capping the paths of the graph as described above. Each line of the table has neutral effect on the distance, being therefore optimal. Moreover, it is easy to see that it is not possible to obtain a "better" capping, with more cycles and/or less caps, that would even decrease the distance. An example of a graph with optimally capped paths is given in Figure 3.2.

Table 3.2: Linking paths from $\mathsf{G}_{\mathrm{R}}(\mathbb{A}, \mathbb{B})$ of canonical genomes. The symbol $\mathring{\alpha}_{\mathbb{A}}$ represents an artificial adjacency in $\mathbb{A}$ and the symbol $\mathring{\alpha}_{\mathbb{B}}$ represents an artificial adjacency in $\mathbb{B}$. Observe that $\Delta \mathrm{d}_{\mathrm{DCJ}} = \Delta n_* - (\Delta(|\mathcal{C}|) + \Delta(2|\mathcal{P}_{\mathbb{AB}}|))$.

	paths	linking cycle	Δn_*	$\Delta(\lvert\mathcal{C}\rvert)$	$\Delta(2\lvert\mathcal{P}_{\mathbb{AB}}\rvert)$	$\Delta \mathrm{d}_{\mathrm{DCJ}}$
1.	$\mathbb{AB}^{\langle \imath \rangle}$	$\mathbb{O}^{\langle \imath+1 \rangle} = \zeta(\mathbb{AB}^{\langle \imath \rangle})$	+0.5	+1	−0.5	0
2.	$\mathbb{AA}^{\langle \imath \rangle} + \mathbb{BB}^{\langle \jmath \rangle}$	$\mathbb{O}^{\langle \imath+\jmath+2 \rangle} = \zeta(\mathbb{AA}^{\langle \imath \rangle}, \mathbb{BB}^{\langle \jmath \rangle})$	+1	+1	0	0
3.	$\mathbb{AA}^{\langle \imath \rangle}$	$\mathbb{O}^{\langle \imath+2 \rangle} = \zeta(\mathbb{AA}^{\langle \imath \rangle}, \mathring{\alpha}_{\mathbb{B}})$	+1	+1	0	0
4.	$\mathbb{BB}^{\langle \jmath \rangle}$	$\mathbb{O}^{\langle \jmath+2 \rangle} = \zeta(\mathbb{BB}^{\langle \jmath \rangle}, \mathring{\alpha}_{\mathbb{A}})$	+1	+1	0	0

Recall that $\kappa(\mathbb{A})+\kappa(\mathbb{B}) = |\mathcal{P}_{\mathbb{AB}}|+|\mathcal{P}_{\mathbb{AA}}|+|\mathcal{P}_{\mathbb{BB}}|$. Now let $a_* = |\kappa(\mathbb{A})-\kappa(\mathbb{B})| = |\,|\mathcal{P}_{\mathbb{AA}}| - |\mathcal{P}_{\mathbb{BB}}|\,|$, $p_* = \max\{\kappa(\mathbb{A}), \kappa(\mathbb{B})\} = \frac{|\mathcal{P}_{\mathbb{AB}}|}{2} + \max\{|\mathcal{P}_{\mathbb{AA}}|, |\mathcal{P}_{\mathbb{BB}}|\}$ and $u = \min\{|\mathcal{P}_{\mathbb{AA}}|, |\mathcal{P}_{\mathbb{BB}}|\}$. An optimal capping of all paths with the minimum number of caps maximizes the use of line (2.) of Table 3.2 and has p_* caps and a_* artificial adjacencies, while an optimal capping with the maximum number of caps uses only lines (1.) , (3.) and (4.) of Table 3.2 and has $p_* + u$ caps and $a_* + 2u$ artificial adjacencies.

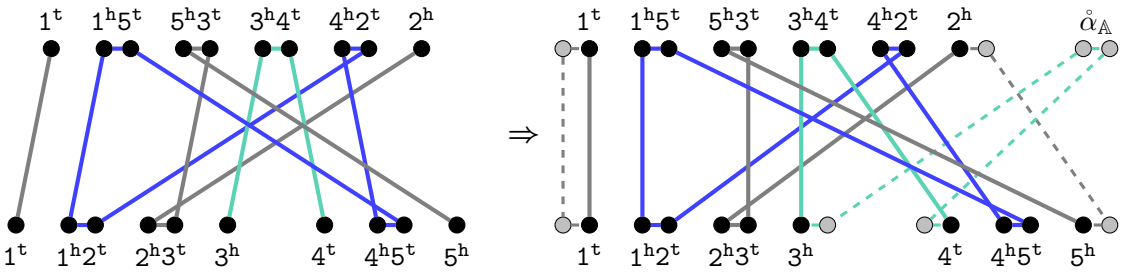

Figure 3.2: Optimal capping of the paths of a relational graph. Cap vertices are drawn in gray.