5. The Double-Cut-and-Join (DCJ) Model

Literature:

Bergeron, A., Mixtacki, J., & Stoye, J. (2006). *A unifying view of genome rearrangements.* Proceedings of WABI 2006. LNBI, Vol. 4175, pp. 163–173

5.1. The double cut and join operation

Definition 18. *Let $G = (V, E)$ be a graph whose vertices have degree 1 or 2. We denote a vertex by its set of* incident edges*, i.e., a vertex incident to edges $e, f \in E$ is denoted by $\{e, f\}$. Likewise, a vertex that is incident to a single edge $g \in E$ is denoted by $\{g\}$.*

Definition 19. *The* double cut and join *(DCJ) operation acts on two vertices u and v of a graph with vertices of degree one or two in one of the following three ways:*

(a) If both $u = \{p, q\}$ and $v = \{r, s\}$ are internal vertices, these are replaced by the two vertices $\{p, r\}$ and $\{s, q\}$ or by the two vertices $\{p, s\}$ and $\{q, r\}$.

(b) If $u = \{p, q\}$ is internal and $v = \{r\}$ is external, these are replaced by $\{p, r\}$ and $\{q\}$ or by $\{q, r\}$ and $\{p\}$.

(c) If both $u = \{q\}$ and $v = \{r\}$ are external, these are replaced by $\{q, r\}$.

In addition, as an inverse of case (c), a single internal vertex $\{q, r\}$ can be replaced by two external vertices $\{q\}$ and $\{r\}$.

Definition 20 (Genome Graph)**.** *The* genome graph *is a graph in which edges correspond to genes (a gene is a tuple of its extremities "*h*" (head), "*t*" (tail)) and vertices correspond to* adjacencies*, i.e., the neighboring extremities of adjacent genes. Each connected component corresponds to a linear or a circular chromosome.*

Note:

- Each gene participates in exactly two adjacencies.
- A vertex corresponding to a single a extremity is called a *telomere*.

Example 16. $(\circ\ 1\ -2\ -3\ 4\ -5\ \circ)\ (\circ\ -6\ 7\ \circ)$ *corresponds to genome graph $G = (V, E)$ with vertex set $V = \{\{1^{\mathrm{t}}\}, \{1^{\mathrm{h}}, 2^{\mathrm{h}}\}, \{2^{\mathrm{t}}, 3^{\mathrm{h}}\}, \{3^{\mathrm{t}}, 4^{\mathrm{t}}\}, \{4^{\mathrm{h}}, 5^{\mathrm{h}}\}, \{5^{\mathrm{t}}\}, \{6^{\mathrm{h}}\}, \{6^{\mathrm{t}}, 7^{\mathrm{t}}\}, \{7^{\mathrm{h}}\}\}$ and edge set $E = \{(x^{\mathrm{h}}, x^{\mathrm{t}}) \mid x = 1..7\}$.*

The genome graph can represent multichromosomal genomes constituting linear and/or circular chromosomes with oriented (signed) genes. This makes it more powerful than the genome models that we studied so far. What kind of rearrangements can be modeled by a double-cut-and-join operation?

- Inversion

$$(\circ\ 1\ |\ -2\ \ -6\ |\ 4\ \ -5\ \circ)(\circ\ \ -3\ 7\ \circ)\ \rightarrow\ (\circ\ 1\ 6\ 2\ 4\ \ -5\ \circ)\ (\circ\ \ -3\ 7\ \circ)$$

- Chromosome fission

$$(\circ\ 1\ |\ 6\ 2\ |\ 4 - 5\ \circ)\ (\circ\ \ -3\ 7\ \circ)\ \rightarrow\ (\circ\ 1\ 4\ \ -5\ \circ)\ (6\ 2)\ (\circ\ \ -3\ 7\ \circ)$$

- Chromosome fusion

$$(\circ\ 1\ 4\ \ -5\ |\ \circ)\ (\circ\ |\ \ -3\ 7\ \circ)\ (6\ 2)\ \rightarrow\ (\circ\ 1\ 4\ \ -5\ \ -3\ 7\ \circ)\ (6\ 2),\ \text{discard}\ (\circ\ \circ)$$

- Chromosome linearization

$$(\circ\ 1\ 4\ \ -5\ |\ \circ)\ (6\ |\ 2)\ (\circ\ |\ -3\ 7\ \circ)\ (\circ\ |\ \circ)\ (\circ\ 1\ 4\ \ -5\ \ -3\ 7\ \circ)\ (\circ\ 2\ 6\ \circ)$$

- Chromosome circularization
- Translocation

$$(\circ\ 1\ 4\ \ -5\ \ -3\ |\ 7\ \circ)\ (\circ\ 2\ |\ 6\ \circ)\ \rightarrow\ (\circ\ 1\ 4\ \ -5\ \ -3\ 6\ \circ)\ (\circ\ 2\ 7\ \circ)$$

- Transposition (2 DCJs)

$$\begin{aligned}(\circ\ 1\ |\ 4\ \ -5\ |\ -3\ 6\ \circ)\ (\circ\ 2\ 7\ \circ)\ &\rightarrow\ (\circ\ 1\ \ -3\ |\ 6\ \circ)\ (4\ 5|)\ (\circ\ 2\ 7\ \circ)\\ &\rightarrow\ (\circ\ 1\ \ -3\ 4\ \ -5\ 6\ \circ)\ (\circ\ 2\ 7\ \circ)\end{aligned}$$

- Block interchange (4 DCJs)

5.2. The adjacency graph

In an adjacency graph $AG(A, B)$ of genomes A and B, vertices are the adjacencies and telomeres of A and B and edges connect corresponding extremities of A and B.

More formally, we have:

Definition 21 (Adjacency Graph)**.** *The* adjacency graph $AG(A, B)$ *of two genome graphs A and B is an undirected multi-graph whose set of vertices are the elements of the multi-set $V(A) \cup V(B)$ and for each $v \in V(A)$ and $u \in V(B)$ for which $u \cap v \neq \emptyset$ there is an edge between u and v in $AG(A, B)$.*

The adjacency graph is a graph whose vertices have again degree 1 or 2, hence we can apply the same DCJ operation on it as on the genome graph. *But:* we will only apply DCJ operations on vertices associated with genome A.

Observation 10. *The connected components of the adjacency graph are:*

- *Cycles,*
- *paths of even length (between two linear chromosomes), and*
- *paths of odd length (between a linear and a circular chromosome).*

Observation 11. *Cycles of length 2 correspond to common adjacencies.*

Observation 12. *Paths of length 1 correspond to common telomeres*

Observation 13. *The adjacency graph between two identical genomes consists of cycles of length 2 and paths of length 1.*

5.3. Sorting by DCJs

Problem 4 (Sorting by DCJs)**.** *Given the genome graph $AG(A, B)$ of two genomes A and B, find a minimum number of DCJ operations $O_1, O_2, \ldots, O_d$ to transform A into B. We call $dcj(A, B) := d$ the* double-cut-and-join *distance.*

Lemma 4. *Two genomes with N genes are identical if their adjacency graph has $N - \frac{I}{2}$ cycles, where I is the number of odd paths.*

In other words, $N = C + \frac{I}{2} \Leftrightarrow N - C - \frac{I}{2} = 0$, where $C :=$ #cycles and $I :=$ #odd paths. The application of one DCJ operation in the graph $AG(A, B)$ can change

- the number of cycles by -1, 0 or $+1$, or
- number of odd paths by -2, 0 or $+2$.
- No DCJ changes odd paths and cycles at the same time.

Therefore, we have $\Delta C + \frac{I}{2} = -1, 0, +1$. This directly leads to a lower bound of the DCJ distance: $dcj(A, B) \geq N - C - \frac{I}{2}$.

Let us look at this simple greedy sorting algorithm:

In each step, $C + \frac{I}{2}$ is increased by one, therefore the greedy algorithm transforms A into B in $N - C - \frac{I}{2}$ steps, which is the lower bound, i.e., Algorithm 2 is optimal. Using tables to relate between gene extremities and vertices of the adjacency graph, the algorithm runs in $O(N)$ time and space.

Algorithm 2 Greedy sorting by DCJ

Input: Adjacency graph $AG(A, B)$

Output: Sequence $O_1, ...O_d$ of DCJ operations such that $d = dcj(A, B)$

1: **for each** adjacency $\{p, q\}$ in genome B **do**
2: let u, v be the elements of genome A that contain p and q, respectively
3: **if** $u \neq v$ **then**
4: replace u and v in A by $\{p, q\}$ and $(u \backslash \{p\}) \cup (v \backslash \{q\})$ and report the corresponding DCJ operation
5: **end if**
6: **end for**
7: **for each** telomere $\{p\}$ in genome B **do**
8: let u be the element of genome A that contains p
9: **if** u is an adjacency **then**
10: replace u in A by $\{p\}$ and $(u \setminus \{p\})$ and report the corresponding DCJ operation
11: **end if**
12: **end for**