Algorithms in Genome Research

Pedro Feijao

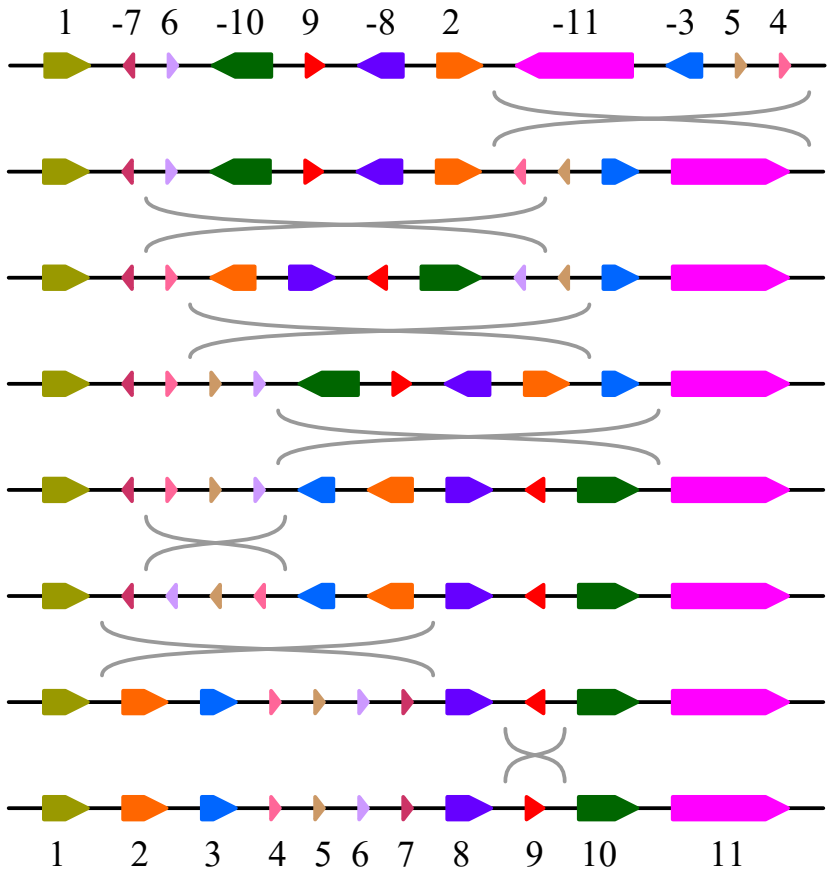
Summer 2014

pfeijao@cebitec.uni-bielefeld.de

Multiple Genome Rearrangement and Breakpoint Models

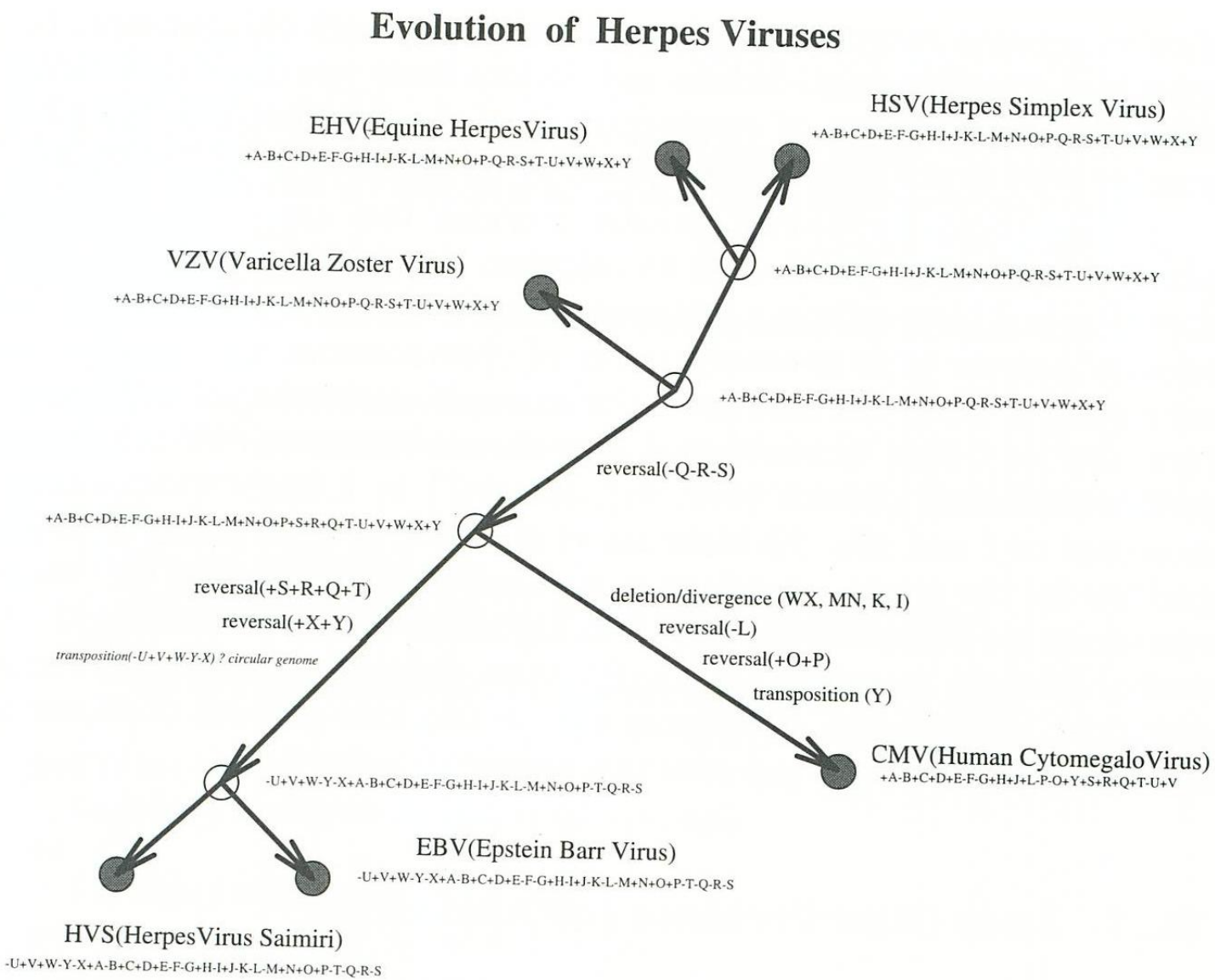
Genome Rearrangement Scenarios

- Finding genome rearrangement scenarios between two genomes is usually easy.



Genome Rearrangement Scenarios

- What if we have more genomes? Can we find an evolutionary scenario?
- Ideally, we want a **rearrangement phylogeny**, explaining ancestral configurations and rearrangement scenarios.
- For instance, something like:



Pevzner, Computational Molecular Biology: An Algorithmic Approach (2000)

Multiple Genome Rearrangement

- The complexity of many combinatorial problems increases when the number of objects increase from 2 to 3.
- Genome Rearrangement is no exception: when comparing 3 (or more) genomes, most rearrangement models are NP-hard.

Multiple Genome Rearrangement

- We are looking for the *most parsimonious phylogenetic tree*. More formally:

Multiple Genome Rearrangement Problem – MGR

Given n genomes, find a tree T with the n genomes as *leaf nodes* and assign ancestral genomes to internal nodes of T such that the tree is optimal, i.e., the sum of rearrangement distances over all edges of the tree is minimal.

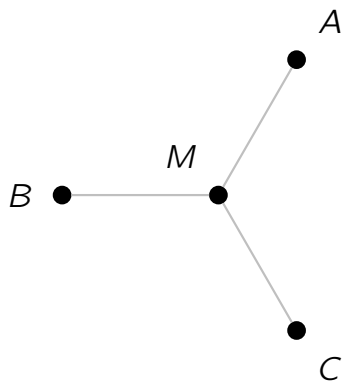
- This problem is also called the **Big Parsimony Problem**.
- In the **Small Parsimony Problem**, a tree T is given, and only the ancestral assignment is needed.
- The simplest form of the MGR is the **median problem**, when three input genomes are considered.

Genome Median Problem

Given three genomes A, B and C, and a genome distance measure d, find a genome M where the **median score**

$$s(M) = d(A, M) + d(B, M) + d(C, M)$$

is minimized.



This can be used as a subproblem to solve the Small Parsimony, iteratively finding the median in the internal nodes of the tree until convergence is achieved.

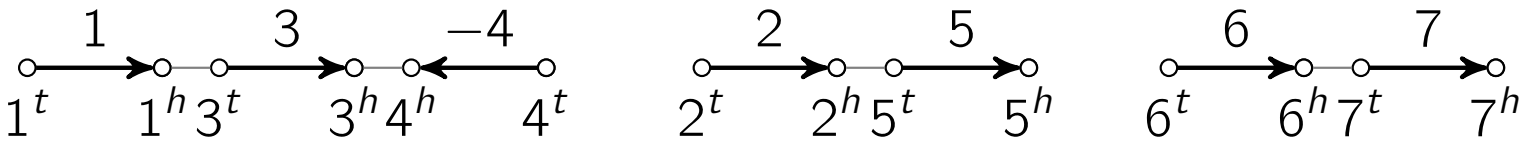
Genome Median Problem

Unfortunately, the median problem is NP-hard for most rearrangement distances, except for *breakpoint distances* in some cases.

- **Unichromosomal BP**: NP-hard
  - Linear Genomes: Pe'er and Shamir, 1998
  - Circular Genomes: Bryant, 1998
- **Reversal**: NP-hard (Caprara, 1997)
- **DCJ**: NP-hard (Caprara, 1997; Tannier et al. 2009)
- **Multichromosomal BP**: $O(n^3)$ (Tannier et al. 2009); $O(n\sqrt{n})$ (Kováč, 2013)
- **Single-Cut-or-Join**: $O(n)$ (Feijão and Meidanis, 2009)

Multichromosomal BP Distance

- Proposed by Tannier et al., in 2009.
- Similarly to the DCJ model, genomes are defined as sets of adjacencies and telomeres, given a gene set $\mathcal{A}$.
- For instance, given $\mathcal{A} = \{1, 2, 3, 4, 5, 6, 7\}$, we can define the genome $A = \{1^t, 1^h3^t, 3^h4^h, 4^t, 2^t, 2^h5^t, 5^h, 6^t, 6^h7^t, 7^h\}$



Multichromosomal BP Distance

Multichromosomal BP Distance – Tannier et al., 2009

Given genomes A and B, the multichromosomal BP distance is defined as

$$d_{\mathrm{BP}}(A, B) = N - A - \frac{T}{2}$$

where N is the number of genes, A is the number of common adjacencies and T the number of common telomeres in A and B.

Alternatively, using the **Adjacency Graph**:

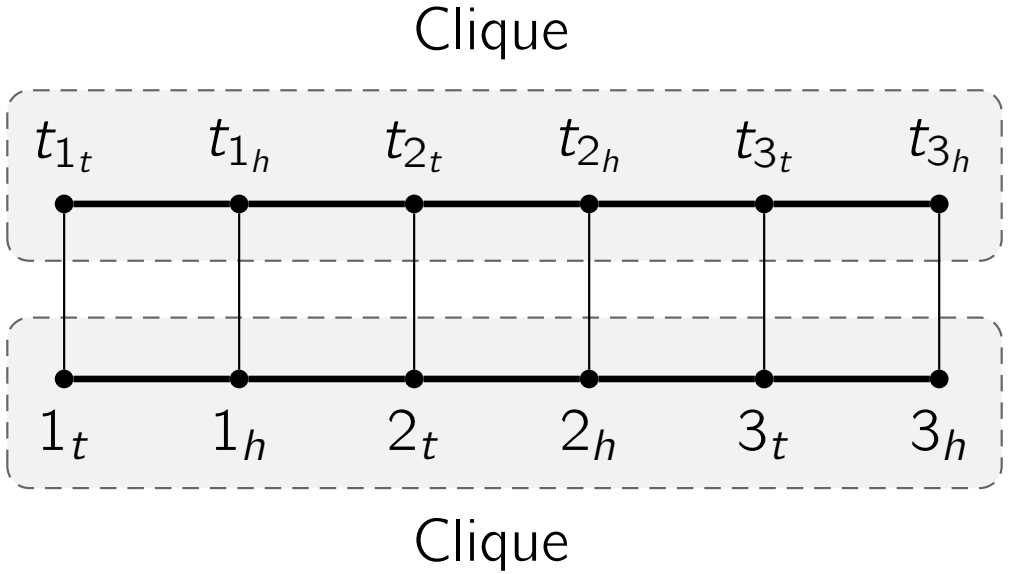
$$d_{\mathrm{BP}}(A, B) = N - C_2 - \frac{P_1}{2}$$

where N is the number of genes, C_2 is the number of cycles of lenght 2 and T the number of paths of lenght 1 in $AG(A, B)$.

Median Problem - BP Distance

- Given a gene set $\mathcal{A}$, consider a graph G whose vertex set has two vertices, x and t_x, for each extremity x of the genes in $\mathcal{A}$.
- There is an edge between x and t_x, for all extremities x, and also and edge between **all** pairs of x vertices and all pairs of t_x vertices.

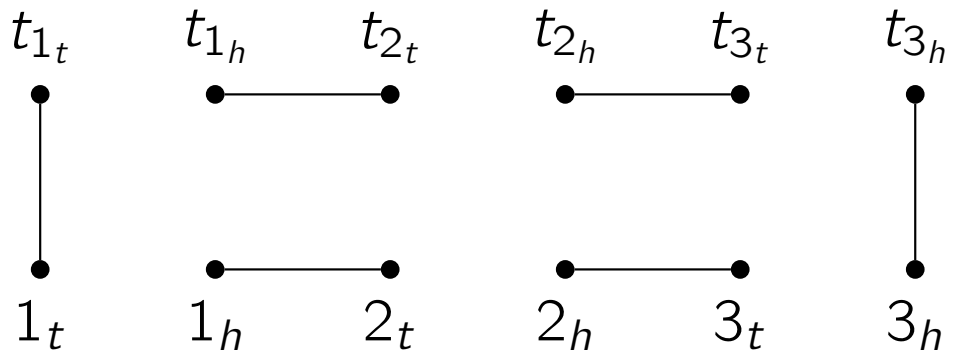
For instance, for $\mathcal{A} = \{1, 2, 3\}$ we have this graph:



Property: **Perfect Matching** in $G \iff$ **Genome** in $\mathcal{A}$.

Example

For gene set $\mathcal{A} = \{1, 2, 3\}$, and genome A = $\{1_t, 1_h2_t, 2_h3_t, 3_h\}$ we have the following matching:



- "Horizontal edges" $\rightarrow$ Adjacencies in the genome.
- "Vertical edges" $\rightarrow$ Telomeres in the genome.

Median Problem - BP Distance

Now consider the same graph G, in an weighted form: Given genomes A, B and C, assign weights to the edges of G in this form:

- **Adjacency weights**: for each adjacency edge (x, y), the weight is # of genomes that have adjacency xy ($w = 0, 1, 2$ or 3).
- **Telomere weights**: for each telomere edge (x, t_x), weight is # of genomes that have telomere x divided by 2 ($w = 0, 1/2, 1$ or $3/2$).
- Any other edge has weight 0.

Matching Weight and Median Score

Claim

Consider three genomes A, B and C, and the weighted graph G. For any genome M, the corresponding weighted matching in G has total weight

$$w = 3N - (d_{\mathrm{BP}}(A, M) + d_{\mathrm{BP}}(B, M) + d_{\mathrm{BP}}(C, M)) = 3N - s(M)$$

where $s(M)$ is the **median score** of M.

Proof?

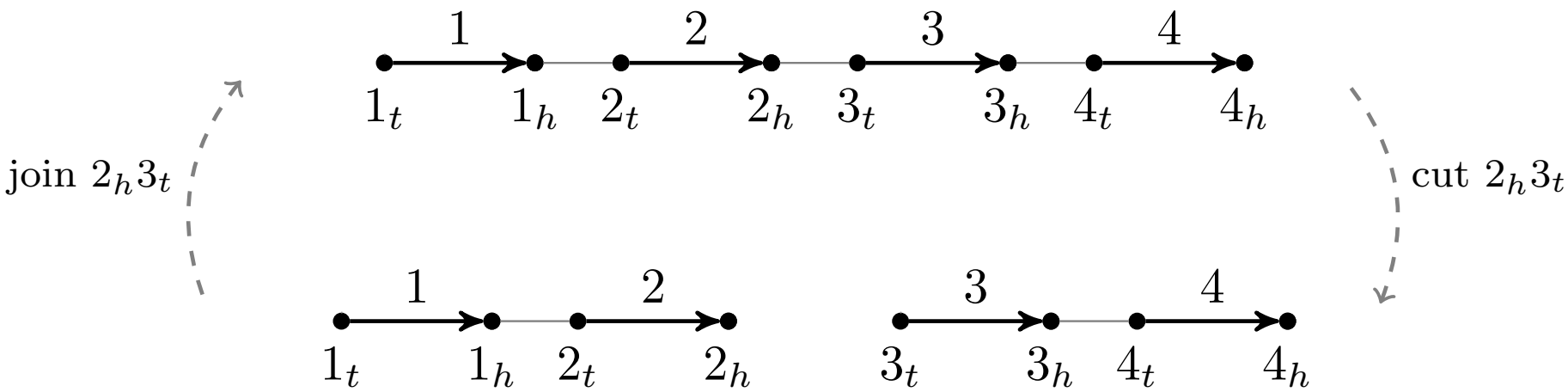
Therefore, solving the **maximum weight perfect matching** problem in G (can be done in $O(n^3)$), we find a median with minimum score, solving the median problem.

Single-Cut-or-Join – SCJ

- Introduced by Feijao and Meidanis in 2009.
- It is very similar to the Multichromosomal BP distance, but slightly simpler.
- The Median problem is solved in $O(n)$. The **small parsimony** problem can also be solved in polynomial time.

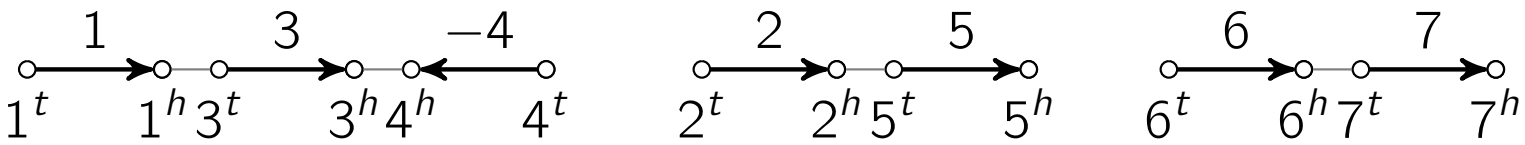
SCJ – Definitions

- A **cut** is an operation that breaks an adjacency in two telomeres.
- A **join** is the reverse operation: two telomeres $\rightarrow$ one adjacency.
- Any single cut **or** single join is a **SCJ**.



Genomes as Sets of Adjacencies

- When a gene set is given, a genome can be uniquely represented as a set of adjacencies, omiting telomeres.
- For instance, given $\mathcal{A} = \{1, 2, 3, 4, 5, 6, 7\}$, we can define the genome $A = \{1_h 3_t, 3_h 4_h, 2_h 5_t, 6_h 7_t\}$

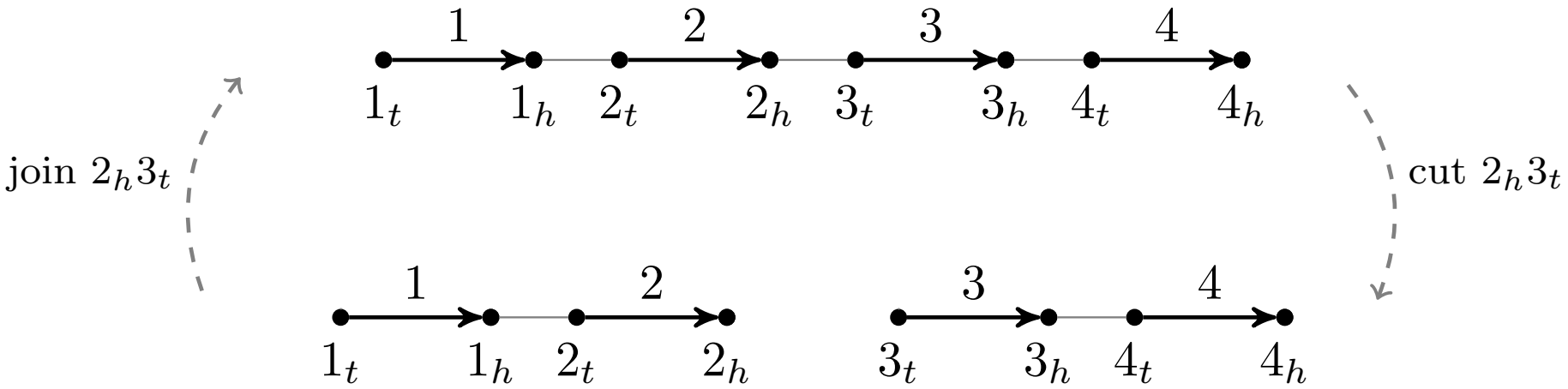


- Then, SCJ operations can be seen as **set operations**:
- A **cut** of an adjacency xy: $A - \{xy\}$.
- A **join** of an adjacency xy : $A \cup \{xy\}$.

Genomes as Sets of Adjacencies - Example

Gene set: $\mathcal{A} = \{1, 2, 3, 4\}$

$$A = \{1_h2_t, 2_h3_t, 3_h4_t\}$$



$$B = \{1_h2_t, 3_h4_t\}$$

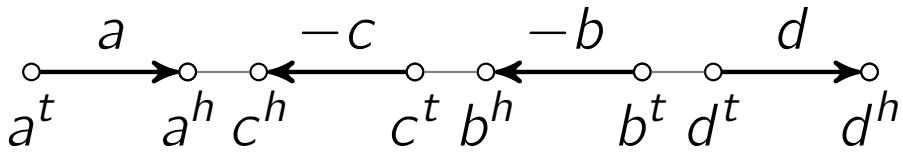
SCJ Distance and Sorting

- How many SCJs do we need to tranform one genome into another?
- If I have two sets A and B, and the only allowed operation is to remove or include elements from the sets, how can I transform A into B in the minimum number of operations?
- One way: First, remove all elements of A that are not present in B.
- Then, include in A all elements of B that are not already in A.
- In set theory: remove $(A - B)$ and include $(B - A)$.
- SCJ: Apply **cuts** of $(A - B)$ and **joins** of $(B - A)$.

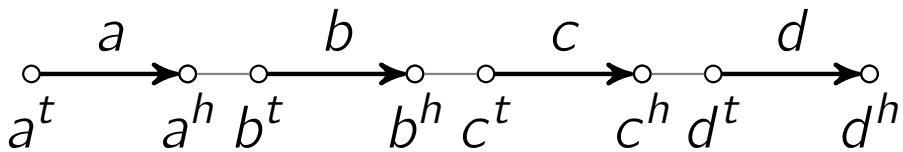
$$d_{\mathrm{SCJ}} = |A - B| + |B - A|$$

SCJ Sorting

$A = \{a_h c_h, c_t b_h, b_t d_t\}$



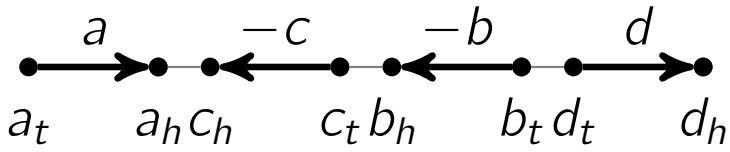
$B = \{a_h b_t, b_h c_t, c_h d_t\}$



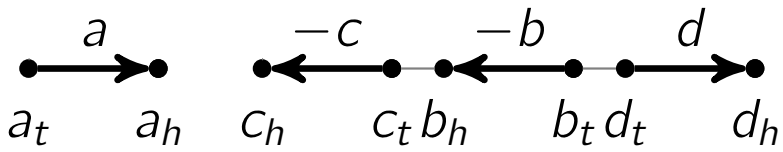
- Red adjacencies must be cut
- Blue adjacencies must be joined

SCJ Sorting

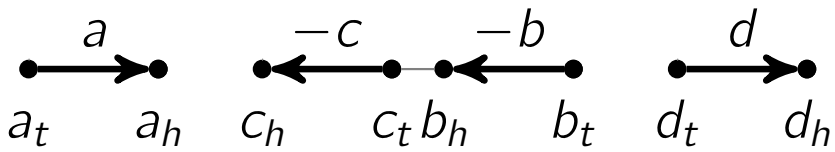
$$A = \{a_h c_h, c_t b_h, b_t d_t\}$$



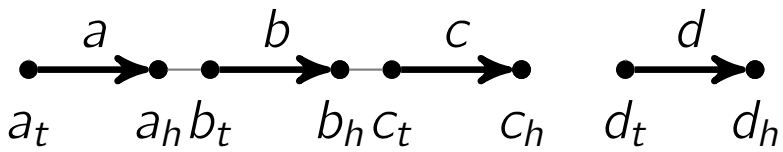
$$A_1 = A - \{a_h c_h\} = \{c_t b_h, b_t d_t\}$$



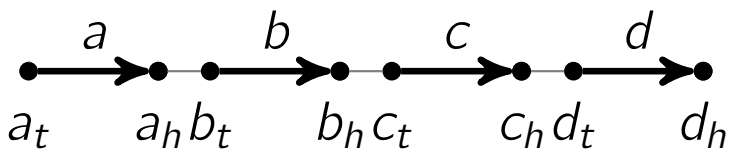
$$A_2 = A_1 - \{b_t d_t\} = \{c_t b_h\}$$



$$A_3 = A_2 \cup \{a_h b_t\} = \{a_h b_t, c_t b_h\}$$



$$A_4 = A_3 \cup \{c_h d_t\} = \{a_h b_t, b_h c_t, c_h d_t\}$$



SCJ Distance with the Adjacency Graph

Simple equation for the SCJ distance using the Adjacency Graph:

$$d_{\mathrm{SCJ}}(A, B) = 2N - 2C_2 - P$$

where N is the number of genes, C_2 and P are the number of cycles of lenght 2 and paths of $AG(A, B)$, respectively.

Proof of SCJ distance by $AG(A, B)$

We know from the definition of SCJ distance and basic set theory that

$$d_{\mathrm{SCJ}}(A, B) = |A - B| + |B - A| = |A| + |B| - 2|A \cap B|.$$

- $|A \cap B|$ = common adjacencies = C_2.
- For any A, we know that $|A| = N - t_A/2$, where t_A is the number of telomeres of A.
- Each path has exactly two telomeres $\Rightarrow P = (t_A + t_B)/2$.

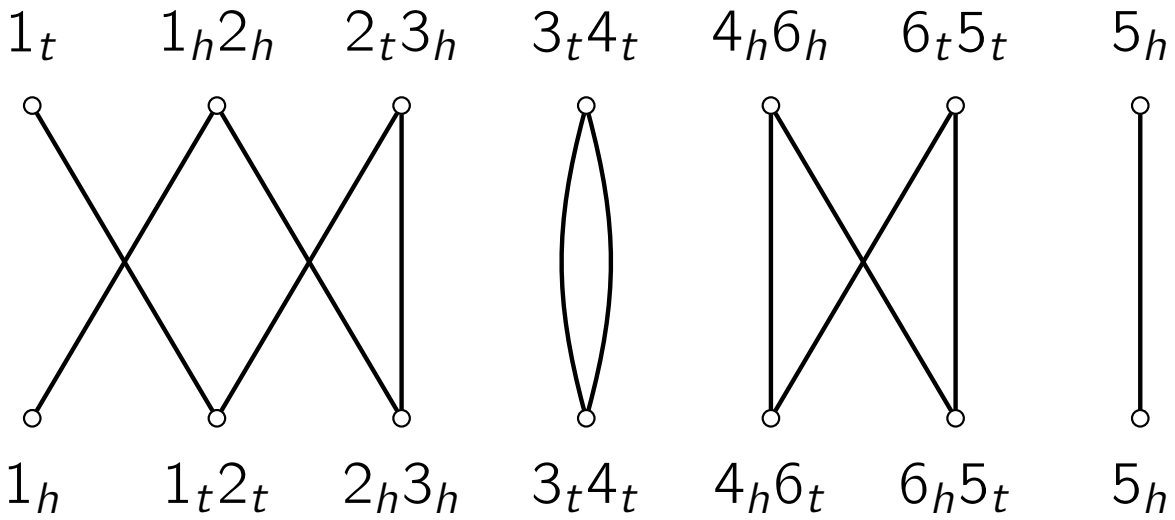
Then,

$$\begin{aligned} d_{\mathrm{SCJ}}(A, B) &= |A| + |B| - 2|A \cap B| \\ &= 2N - (t_A + t_B)/2 - 2C_2 \\ &= 2N - 2C_2 - P. \end{aligned}$$

SCJ with Adjacency Graph – Example

$A = \{1_h2_h, 2_t3_h, 3_t4_t, 4_h6_h, 6_t5_t\}$, $B = \{1_t2_t, 2_h3_h, 3_t4_t, 4_h6_t\}$

$AG(A, B)$



- $d_{\mathrm{SCJ}}(A, B) = |A - B| + |B - A| = 4 + 4 = 8.$
- $d_{\mathrm{SCJ}}(A, B) = 2N - 2C_2 - P = 12 - 2 - 2 = 8.$

Relationship between SCJ, BP and DCJ distances

The "expected" relationship is SCJ = 2BP and SCJ = 4DCJ. The theoretical bounds are:

- Relationship between SCJ and Multichromosomal BP:

$$d_{BP}(A, B) \leq d_{SCJ}(A, B) \leq 2d_{BP}(A, B)$$

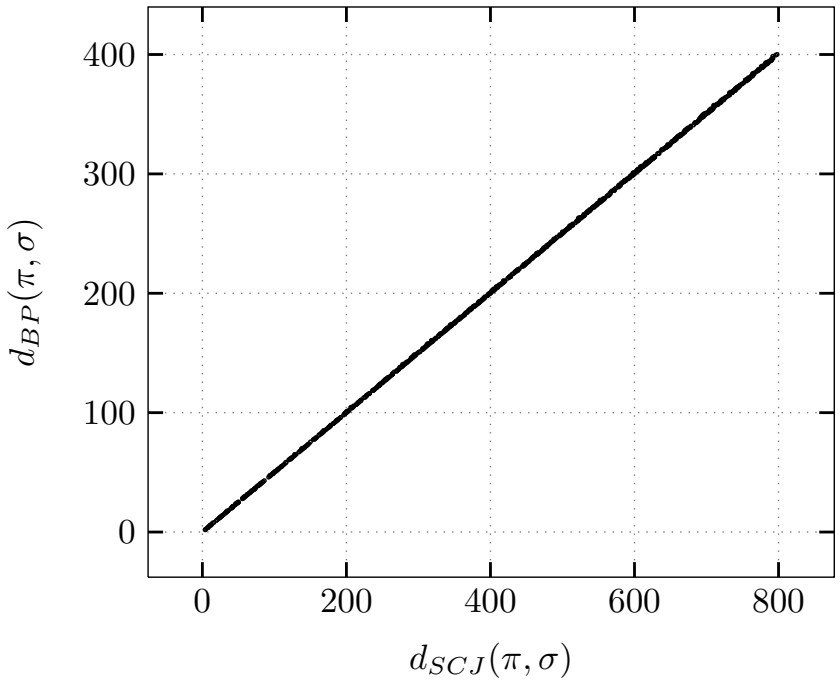
- Relationship between SCJ and DCJ:

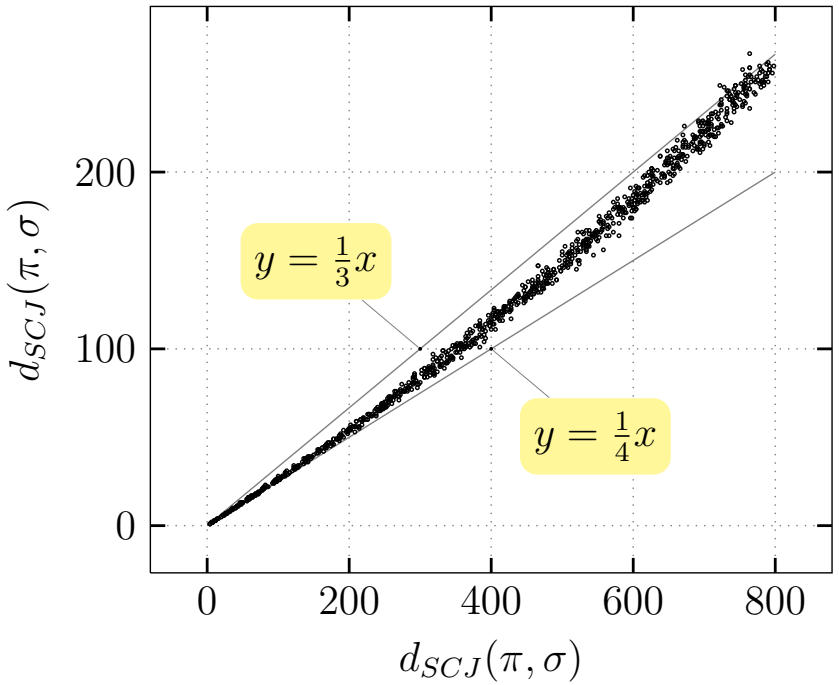
$$d_{DCJ}(A, B) \leq d_{SCJ}(A, B) \leq 4d_{DCJ}(A, B)$$

- All the bounds are tight.

Relationship between SCJ, BP and DCJ distances

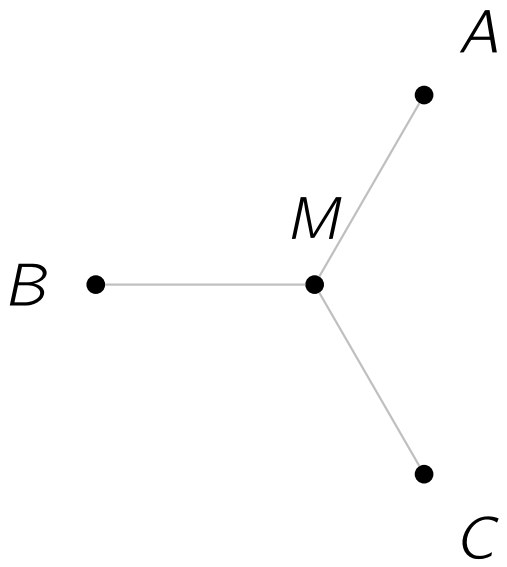
Simulated data:





SCJ Median Problem

- Start with an "empty" genome M and think about the "effect" of adding an adjacency to M.



- If the adjacency is not present in any genome, $\Delta s(M) = +3$.
- If the adjacency is present in 1 genome, $\Delta s(M) = +1$.
- If the adjacency is present in 2 genomes, $\Delta s(M) = -1$.
- If the adjacency is present in 3 genomes, $\Delta s(M) = -3$.
- Adjacencies with $\Delta s(M) < 0$ are **good**.

SCJ Median Problem

Basically, for each adjacency the genomes A, B and C "vote" in favour or against it, depending on whether the adjacency is present or not. The solution is given by

SCJ Median Solution

Given genomes A, B and C, the genome M defined as

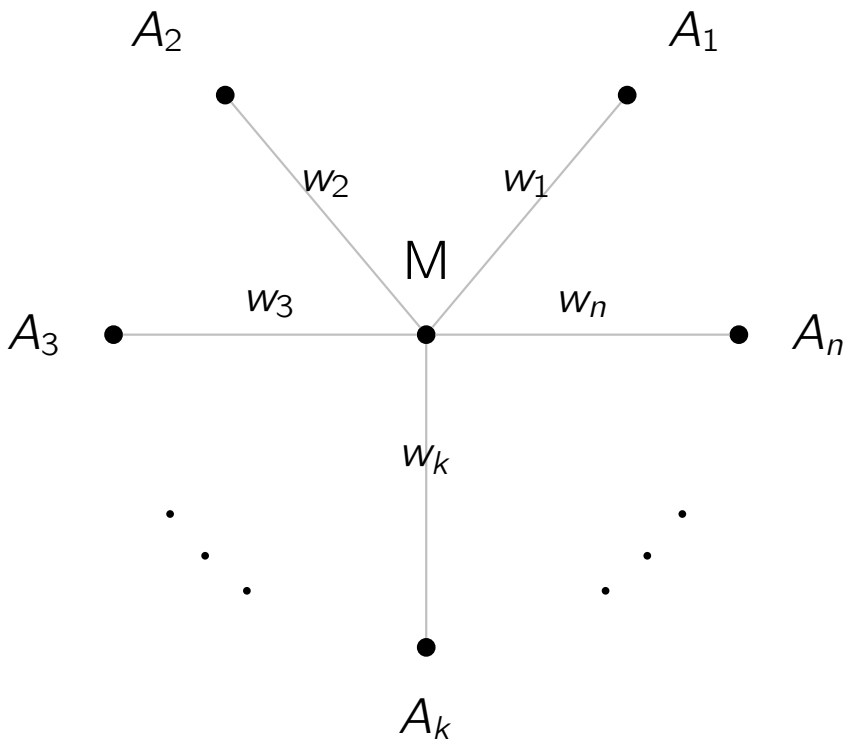
$$M = \{a : \text{adjacency } a \text{ is present in at least two of the input genomes}\}$$

is a median of A, B and C.

Weighted Multiple Genome Median Problem

Formulation

Given n genomes $A_1, \ldots, A_n$ and nonnegative weights $w_1, \ldots, w_n$, find M that minimizes $\sum_{i=1}^{n} w_i \cdot d(A_i, M)$



Weighted Multiple Genome Median Problem

SCJ Solution

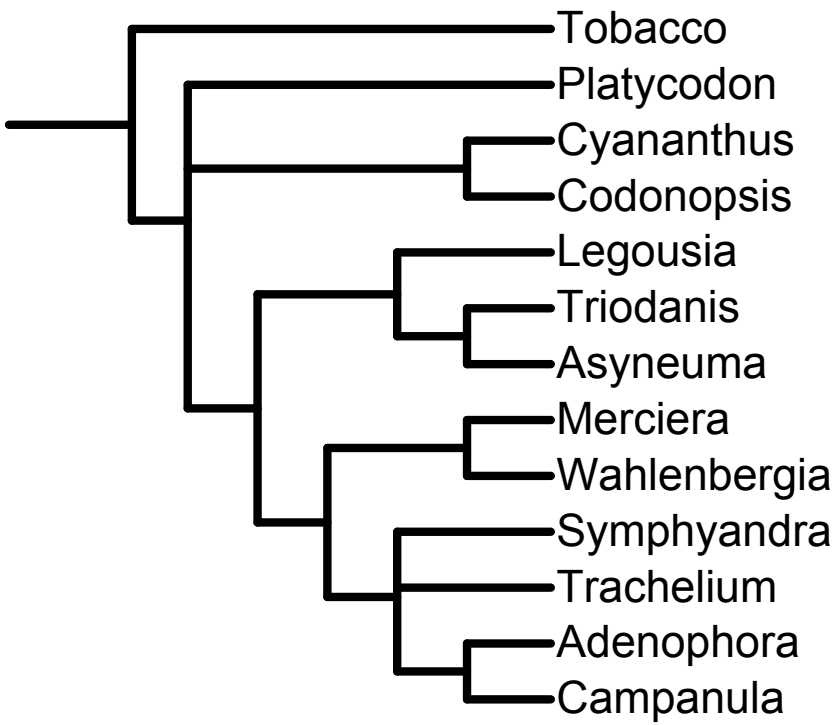
- The genome $M = \{d : f(d) < 0\}$, where

$$f(d) = \sum_{d \notin A_i} w_i - \sum_{d \in A_i} w_i$$

  is a solution to the Weighted Multiple Genome Median Problem.

- If $f(d) \neq 0$ for all adjacencies d, the solution is unique.

The Small Parsimony Problem



Phylogeny for 12 *Campanulaceae* genomes and Tobacco as an outgroup.

- **Small Parsimony Problem**: Assign ancestral genomes the internal nodes of the tree in a way that minimizes the total number of rearrangements in the tree.

The Small Parsimony Problem

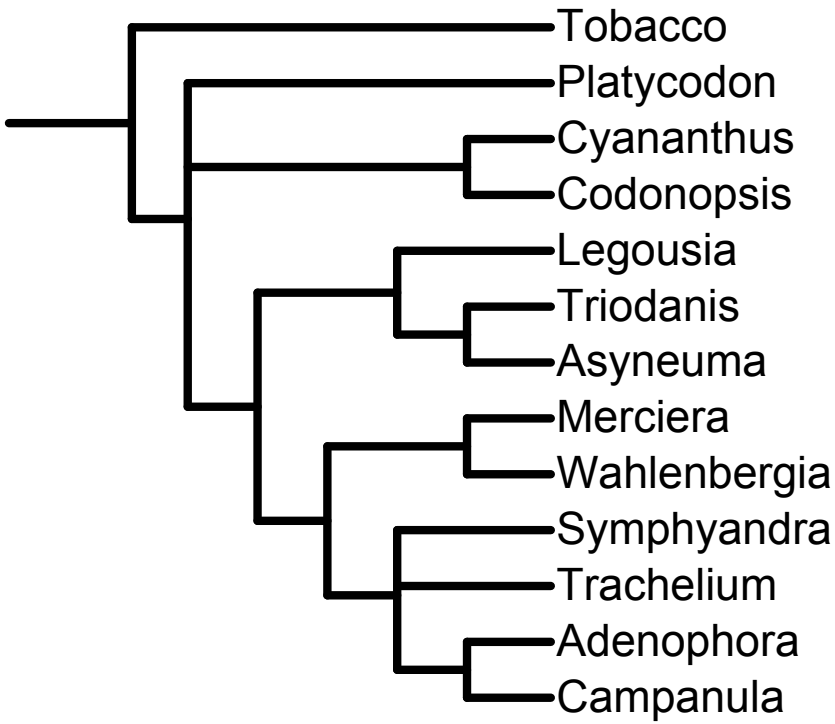
- This problem is NP-hard for any distance where the median is NP-Hard (almost all)
- Also for multichromosomal BP, which median is polynomial, this is NP-Hard (Kováč, 2013).
- The only known polynomial result is with the SCJ distance.

Heuristics for the small parsimony problem

- Sankoff and Blanchette (BPAnalysis, 1997) proposed an iterative procedure: solving median problems in the internal nodes until convergence.
- Also tries to solve the Big Parsimony, by solving the small in all possible trees.
- More recent methods: GRAPPA (Moret et al., 2001); MGR (Bourque and Pevzner, 2002).

Solving the SCJ Small Parsimony

- Fitch's Algorithm (1971) for discrete character sets.



- If each genome is a set of *independent discrete characters*, Fitch's Algorithm finds a tree that minimizes the number of *character changes* in the tree.

SCJ Small Parsimony with Fitch's Algorithm

- Since an adjacency can be seen as a binary character (presence/absence), running Fitch's Algorithm for each adjacency reconstructs ancestral genomes that are **optimal** under the SCJ distance
- The only possible problem is that adjacencies are *not independent*, which could cause conflits, but Feijao and Meidanis (2009) showed how conflicts can be avoided.

Review

- Multiple genome rearrangement problems are usually NP-hard.
- **Median Problem**: Polynomial for Multichromosomal BP and SCJ, NP-hard (or open) for all the rest.
- **Small Parsimony**: Polynomial only for SCJ.