Universität Bielefeld
Technische Fakultät
Genominformatik

Family-Free Genome Comparison

Jens Stoye

Bielefeld University, Germany

SocBiN, 11 June 2014

Comparative genomics

Two levels of genome evolution:

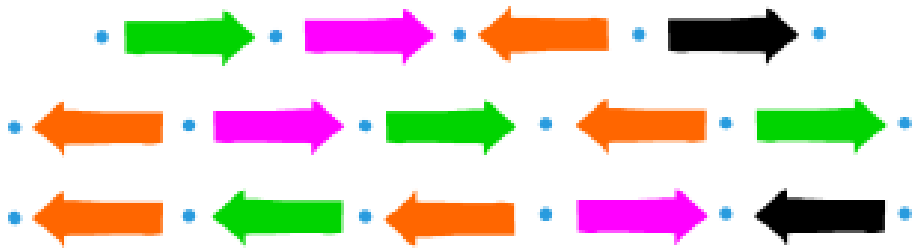
- **Small scale mutations:** point mutations
- **Large scale mutations:** rearrangements, duplications, insertions, deletions

Structural organization provides insights into:

- phylogeny and evolution
- gene function and interactions

Comparative genomics with gene families

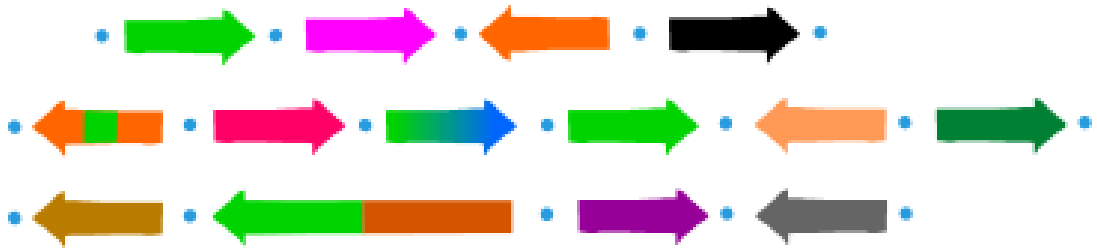
Picture with gene families:



- Simple and powerful data type
- Many databases and tools available
- Produce reasonable results

The Family-free Principle

More realistic picture:

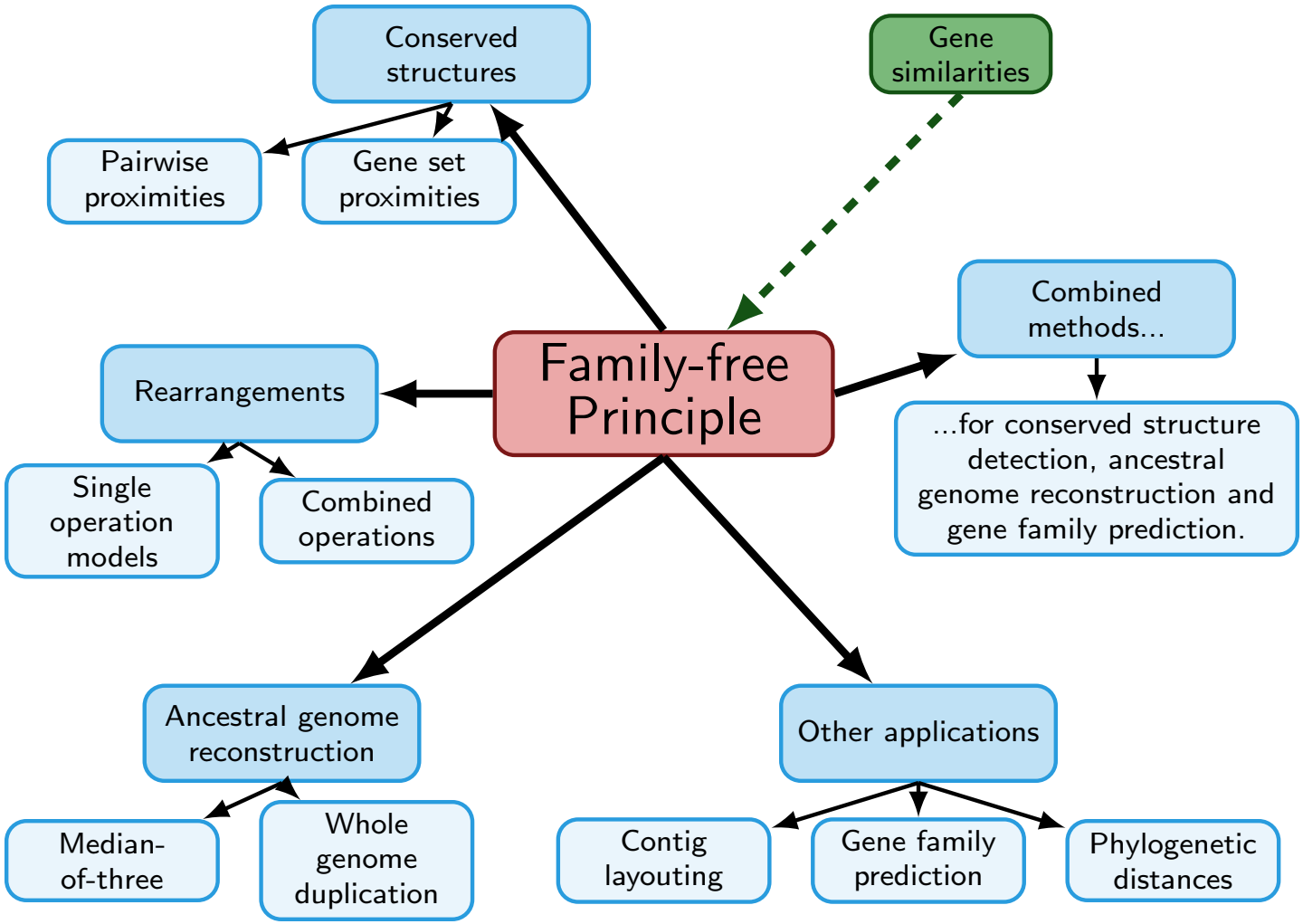


- Computational prediction of gene families is (mostly) unsupervised
- Do not always correspond to biological gene families
- Wrong gene family assignments may produce incorrect results in subsequent analyses

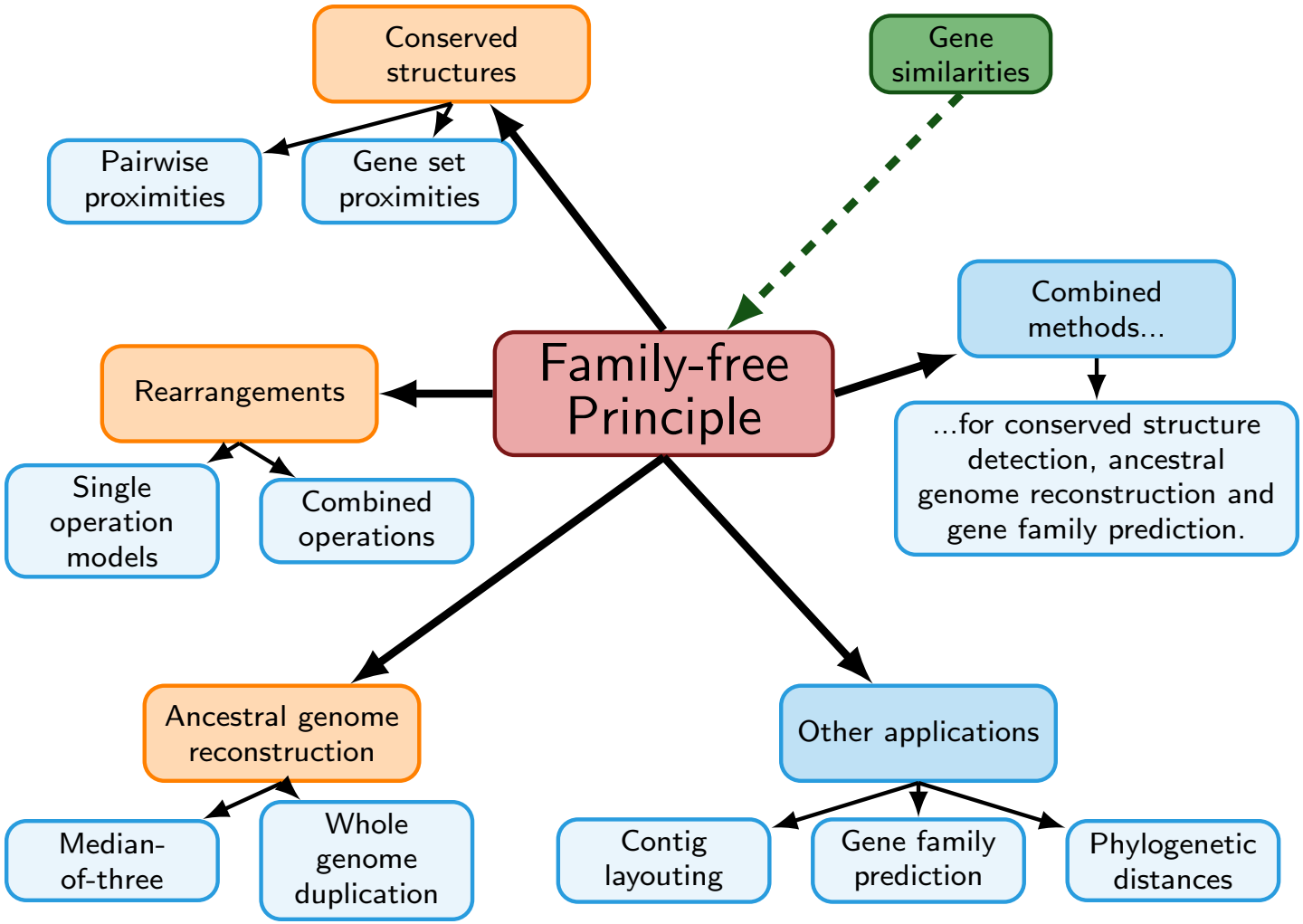
The Family-free Principle

- Gene family assignments not necessary:
  - If subsequent analyses can deal with original data
  - For example gene similarity scores
- We may even invert the scenario:
  - Integrated analysis: ortholog assignments and gene order analysis
  - Gene family assignment based on *positional orthology*

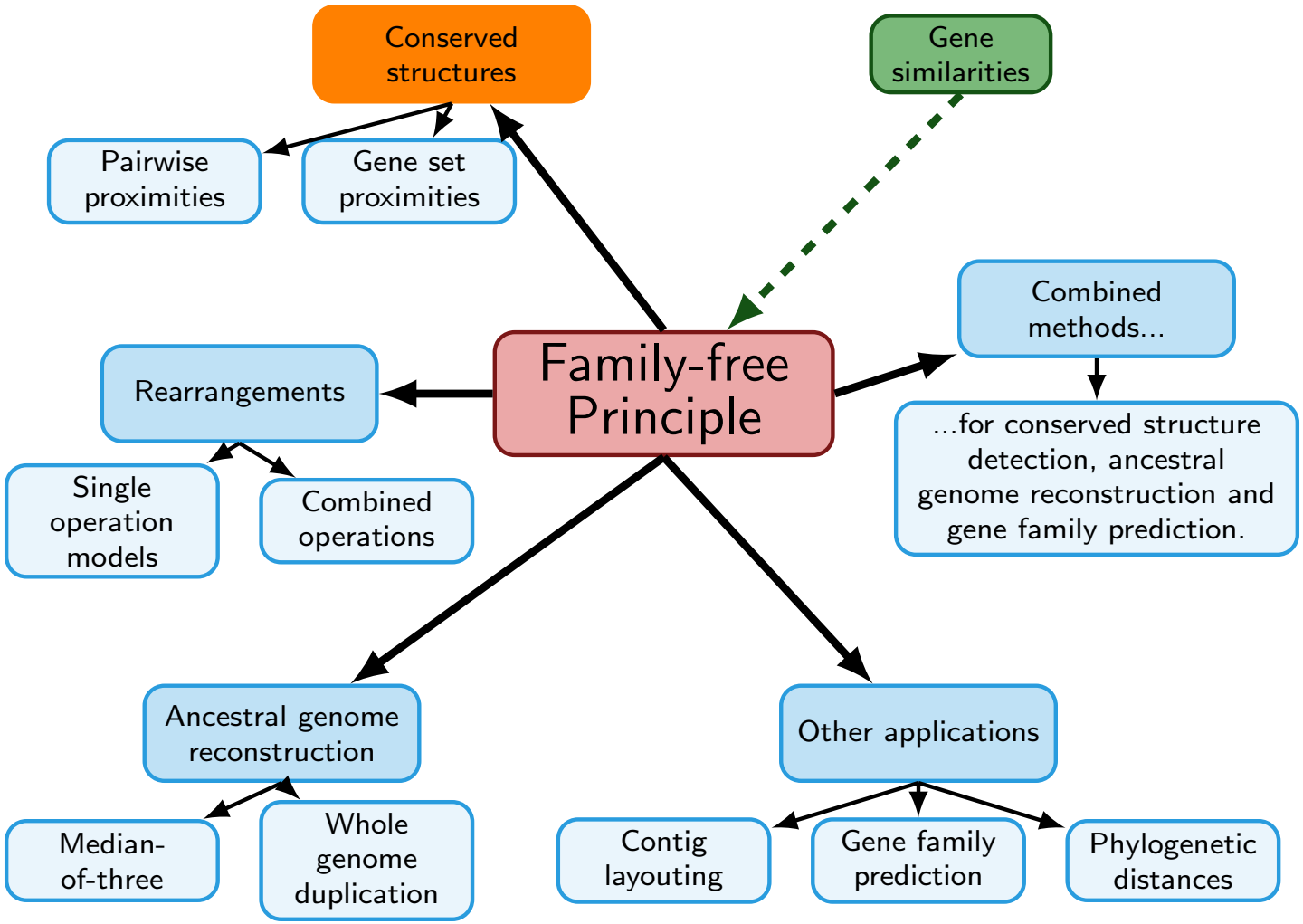
The Family-free Principle



The Family-free Principle

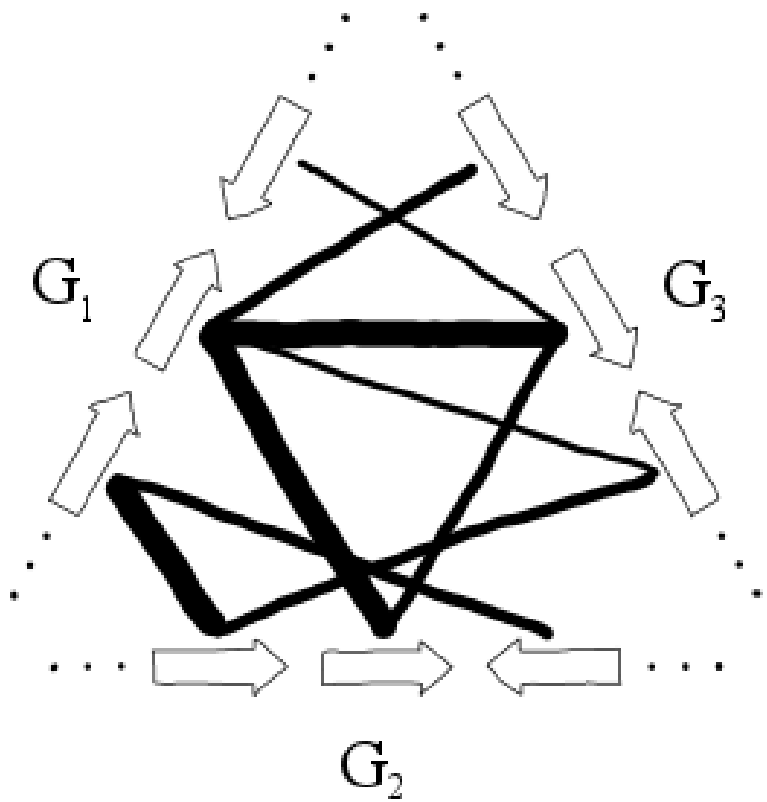


Conserved structures



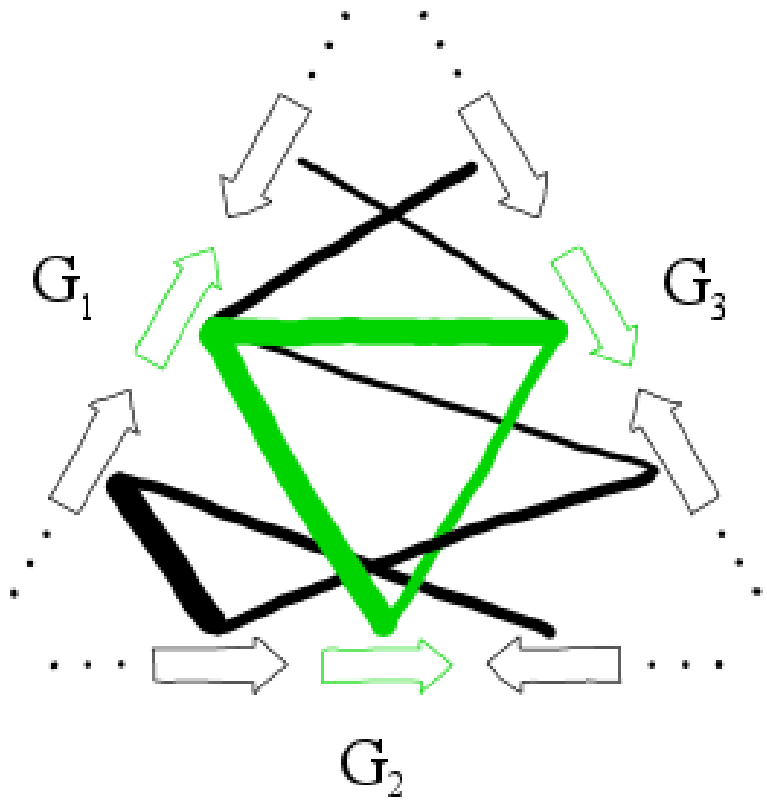
Gene similarity graph

Gene similarity graph of 3 genomes:



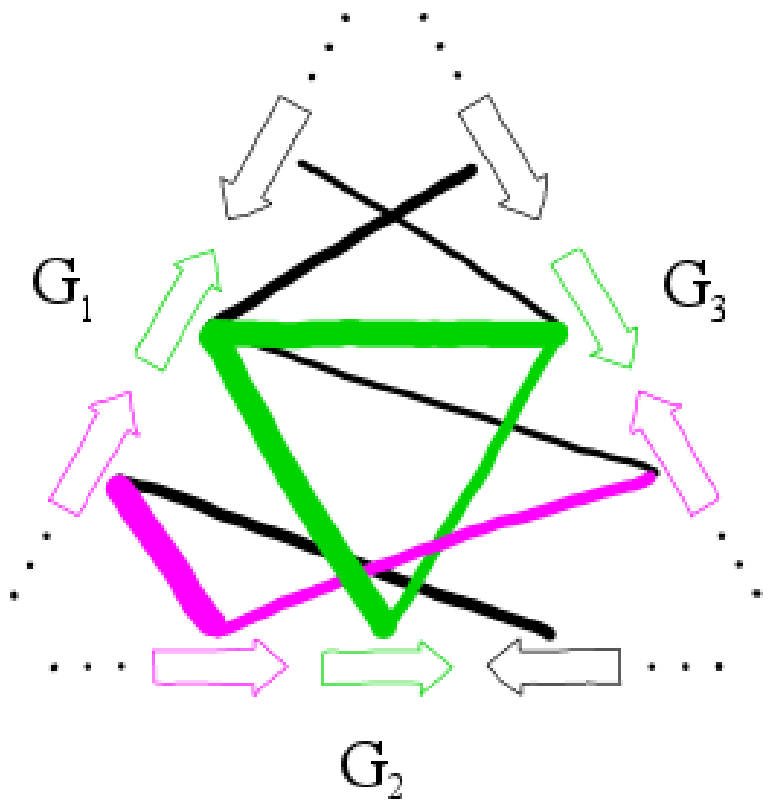
Gene similarity graph

Gene similarity graph of 3 genomes:



Gene similarity graph

Gene similarity graph of 3 genomes:

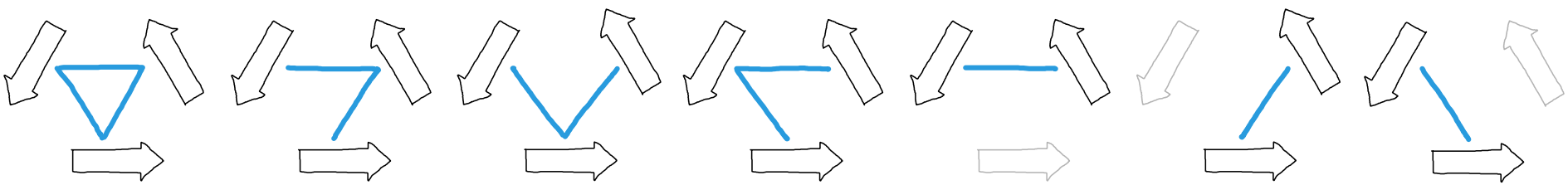


Partial k-matching

Partial k-(dimensional) matching

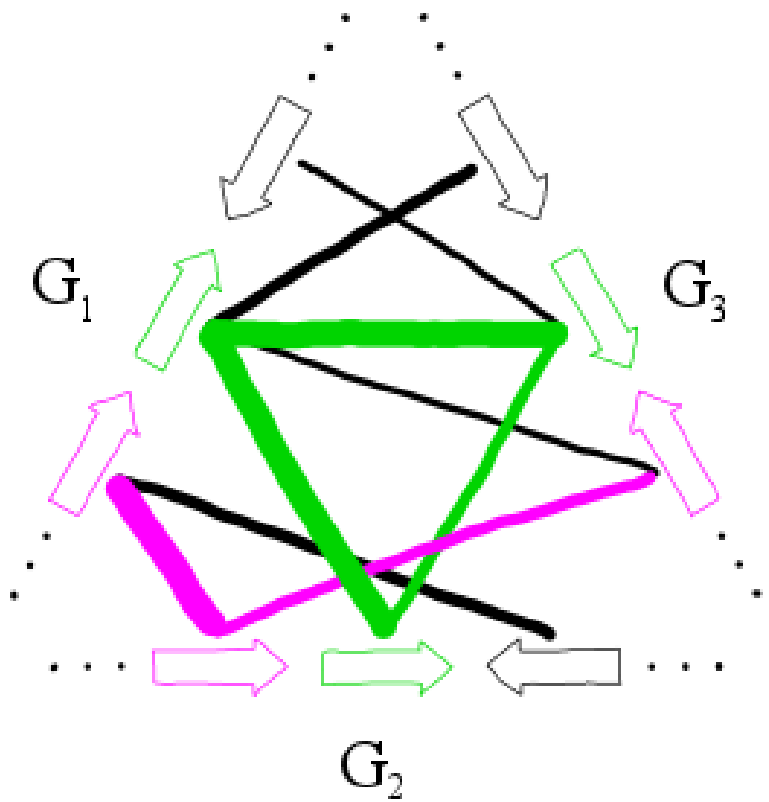
Given a gene similarity graph $B = (G_1, \ldots, G_k, E)$, a *partial k-matching* $\mathcal{M} \subseteq E$ is a selection of edges such that for each connected component $C \subseteq B_{\mathcal{M}} := (G_1, \ldots, G_k, \mathcal{M})$ no two genes in C belong to the same genome.

For $k = 3$: $2^k - 1 = 7$ valid components



Partial k-matching

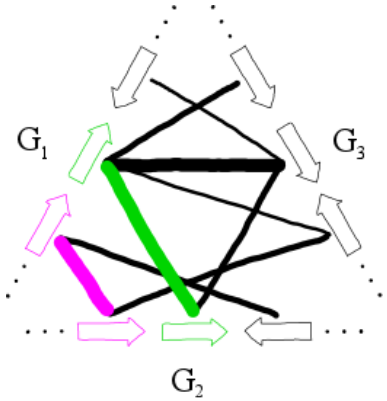
Gene similarity graph of 3 genomes:



. . . how to construct such a matching?

Assessing matching properties

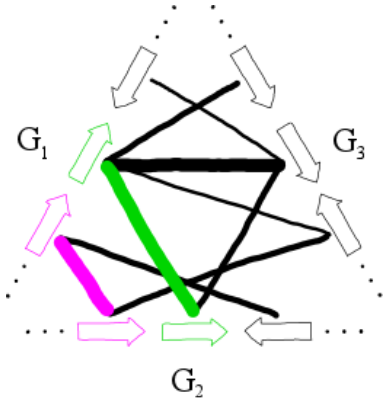
- **Adjacency**: proximity relation between two genes
- Adjacency score for consecutive genes (g, g') in genome G and (h, h') in genome H:



$$s(g, g', h, h') = \begin{cases} \sqrt{w(e_{g,h}) \cdot w(e_{g',h'})} & \text{if } (g, g'), (h, h') \text{ form a conserved adjacency} \\ 0 & \text{otherwise} \end{cases}$$

Assessing matching properties

- **Adjacency**: proximity relation between two genes
- Adjacency score for consecutive genes (g, g') in genome G and (h, h') in genome H:



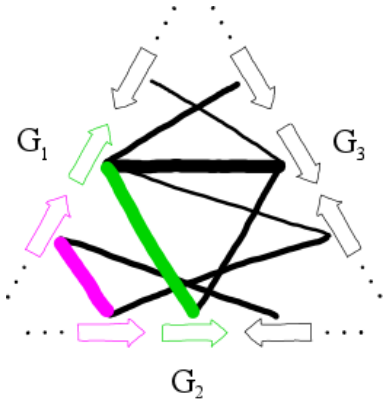
$$s(g, g', h, h') = \begin{cases} \sqrt{w(e_{g,h}) \cdot w(e_{g',h'})} & \text{if } (g, g'), (h, h') \text{ form a conserved adjacency} \\ 0 & \text{otherwise} \end{cases}$$

- **Adjacency measure** in $\mathcal{M}$:

$$adj(\mathcal{M}) = \sum_{G,H} \sum_{\substack{g \text{ left of } g' \text{ in } G \\ h,h' \text{ in } H}} s(g, g', h, h')$$

Assessing matching properties

- **Adjacency**: proximity relation between two genes
- Adjacency score for consecutive genes (g, g') in genome G and (h, h') in genome H:



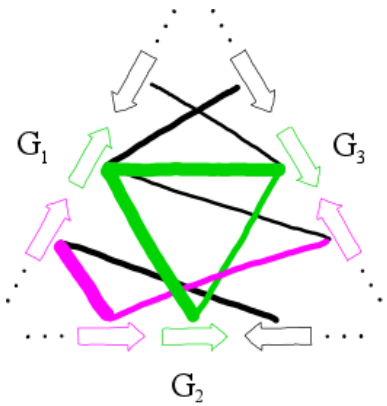
$$s(g, g', h, h') = \begin{cases} \sqrt{w(e_{g,h}) \cdot w(e_{g',h'})} & \text{if } (g, g'), (h, h') \text{ form a conserved adjacency} \\ 0 & \text{otherwise} \end{cases}$$

- **Adjacency measure** in $\mathcal{M}$:

$$adj(\mathcal{M}) = \sum_{G,H} \sum_{\substack{g \text{ left of } g' \text{ in } G \\ h,h' \text{ in } H}} s(g, g', h, h')$$

- **Similarity measure** in $\mathcal{M}$:

$$edg(\mathcal{M}) = \sum_{e \in \mathcal{M}} w(e)$$

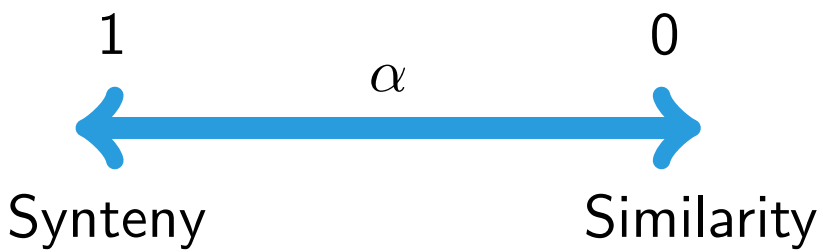


Family-free Adjacencies Problem

Family-free Adjacencies Problem

Find matching $\mathcal{M}$ that maximizes the following formula:

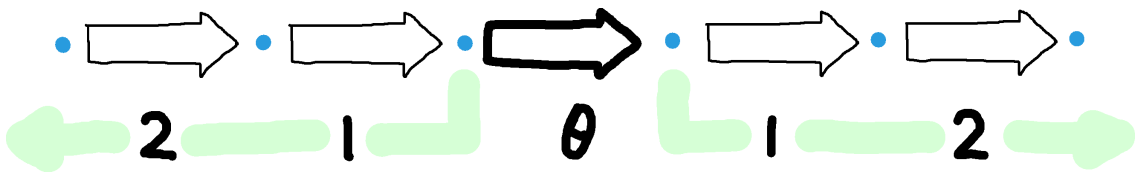
$$\mathcal{F}_\alpha(\mathcal{M}) = \alpha \cdot adj(\mathcal{M}) + (1 - \alpha) \cdot edg(\mathcal{M}) .$$



Gene set proximities: gene clusters

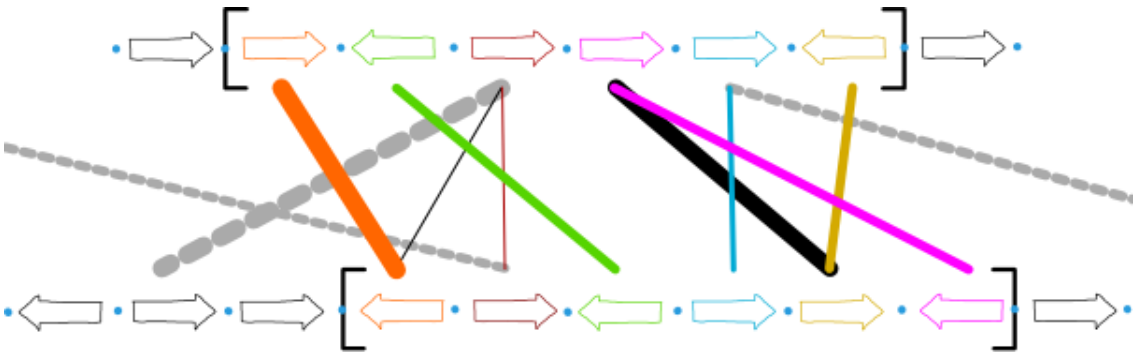
- Relaxation: conserved neighborhood up to $\theta > 0$ genes
- Scoring θ-adjacencies:

$$s^{\theta}(g, g', h, h') = \begin{cases} \sqrt{w(e_{g,h}) \cdot w(e_{g',h'})} & \text{if } (g, g') \text{ and } (h, h') \text{ form a } \theta\text{-adjacency} \\ 0 & \text{otherwise} \end{cases}$$



Gene set proximities: gene clusters

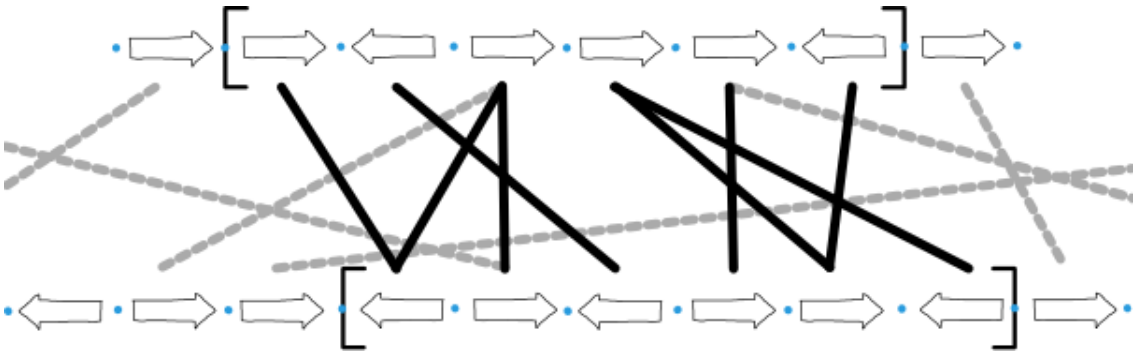
- Based on θ-adjacencies we can define gene clusters as pairs of intervals with large maximum weight matching $\mathcal{M}$:



Gene set proximities: consimilar intervals

Calculating a maximum matching for all pairs of intervals is expensive.

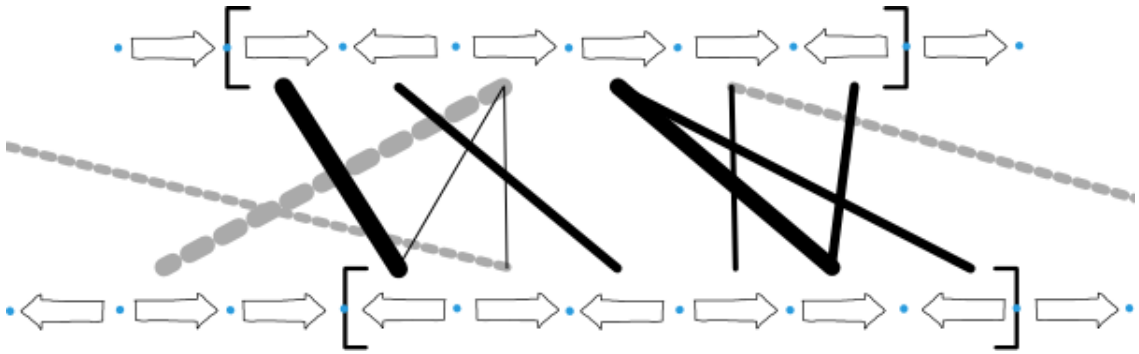
- Therefore use unweighted gene similarity graph
- *Consimilar interval*: many edges inside, no edges to neighbors.
- Algorithm: $O(n^3)$ time



Gene set proximities: consimilar intervals

Calculating a maximum matching for all pairs of intervals is expensive.

- Therefore use unweighted gene similarity graph
- *Consimilar interval*: many edges inside, no edges to neighbors.
- Algorithm: $O(n^3)$ time

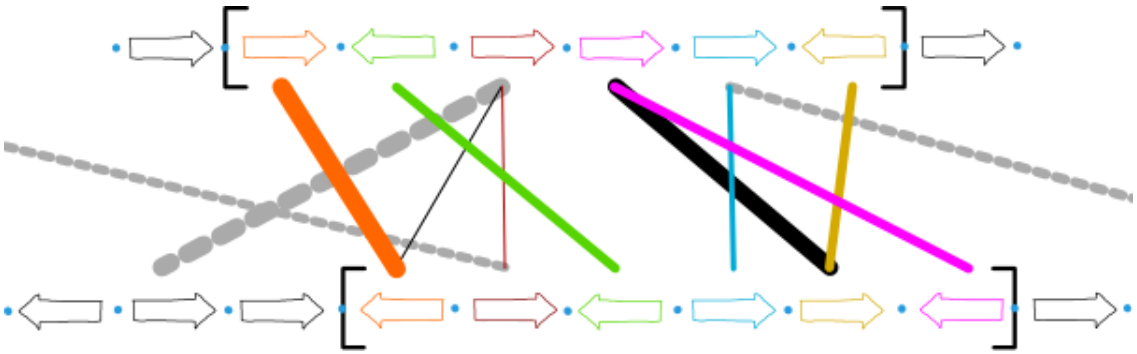


- Ranking by score of maximum weight matching inside the intervals.

Gene set proximities: consimilar intervals

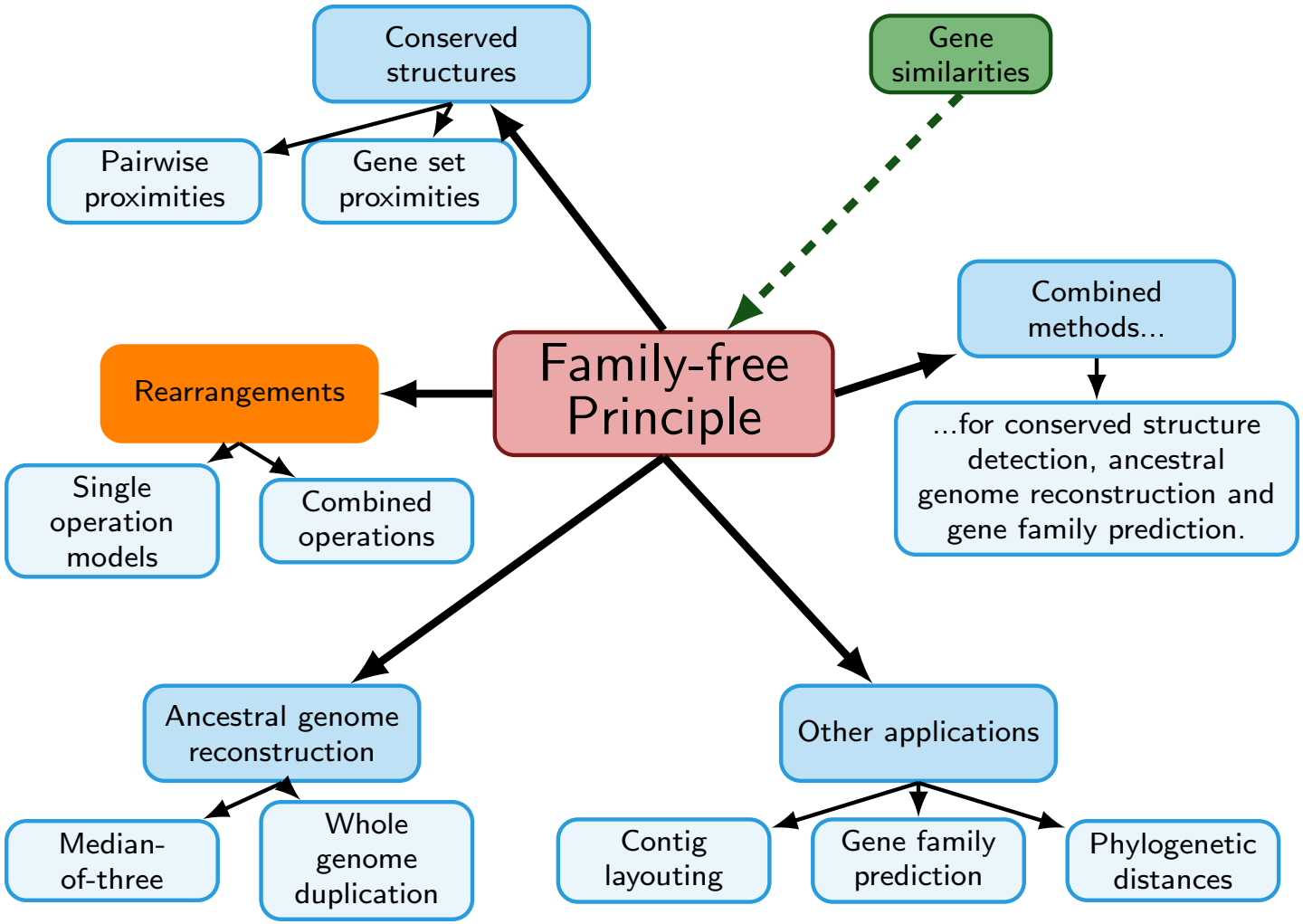
Calculating a maximum matching for all pairs of intervals is expensive.

- Therefore use unweighted gene similarity graph
- *Consimilar interval*: many edges inside, no edges to neighbors.
- Algorithm: $O(n^3)$ time



- Ranking by score of maximum weight matching inside the intervals.

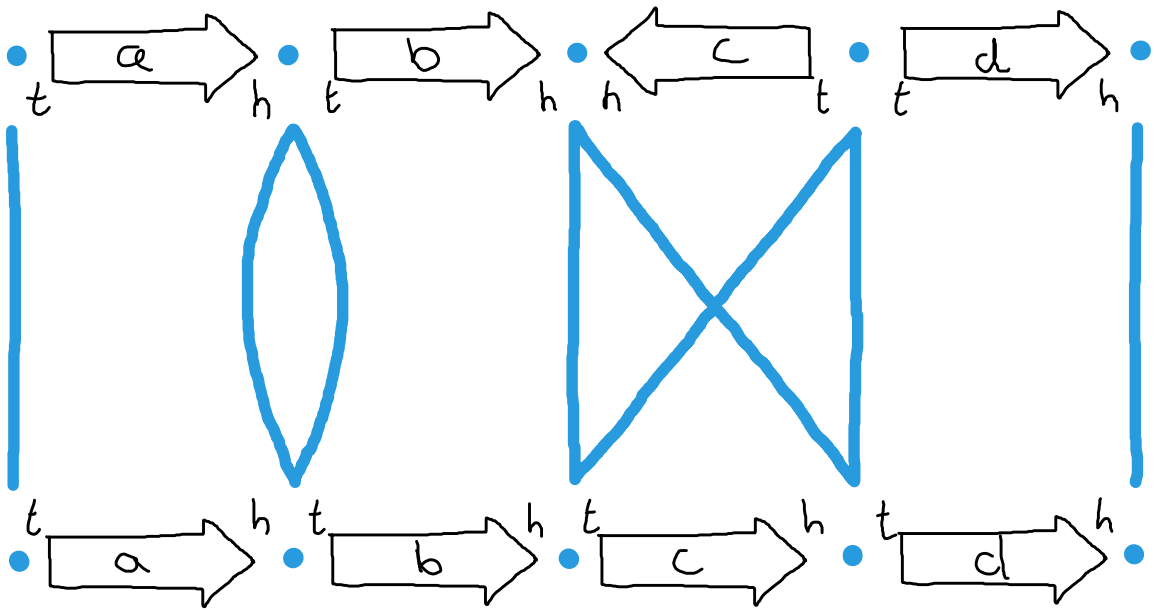
Rearrangements



DCJ – Double Cut and Join

DCJ accounts for rearrangement events: inversion, translocation, fusion, fission, transposition, block interchange

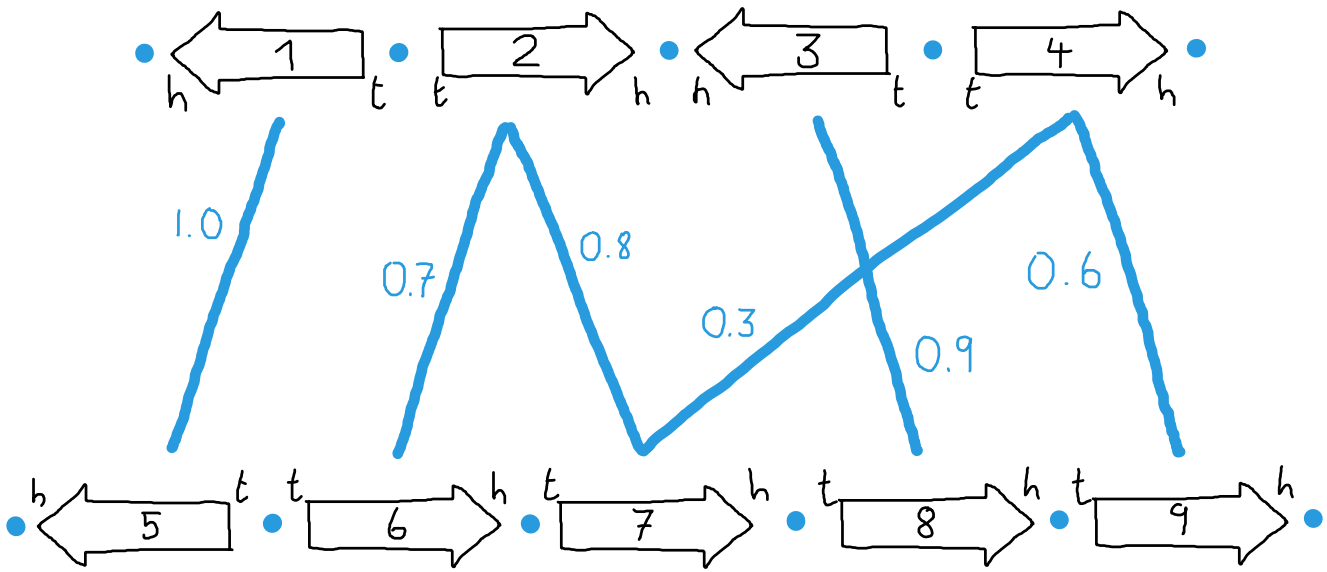
Adjacency graph:



$$\text{distance } d_{DCJ} = N - C - \frac{I}{2}$$

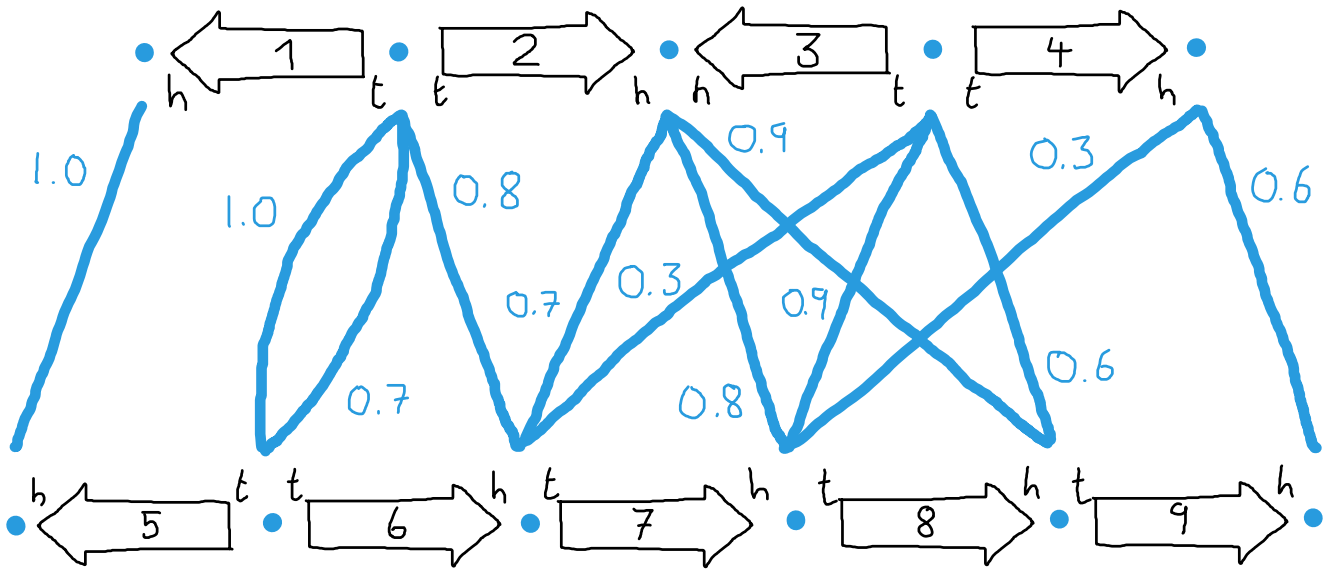
DCJ – Double Cut and Join

From the gene similarity graph . . .



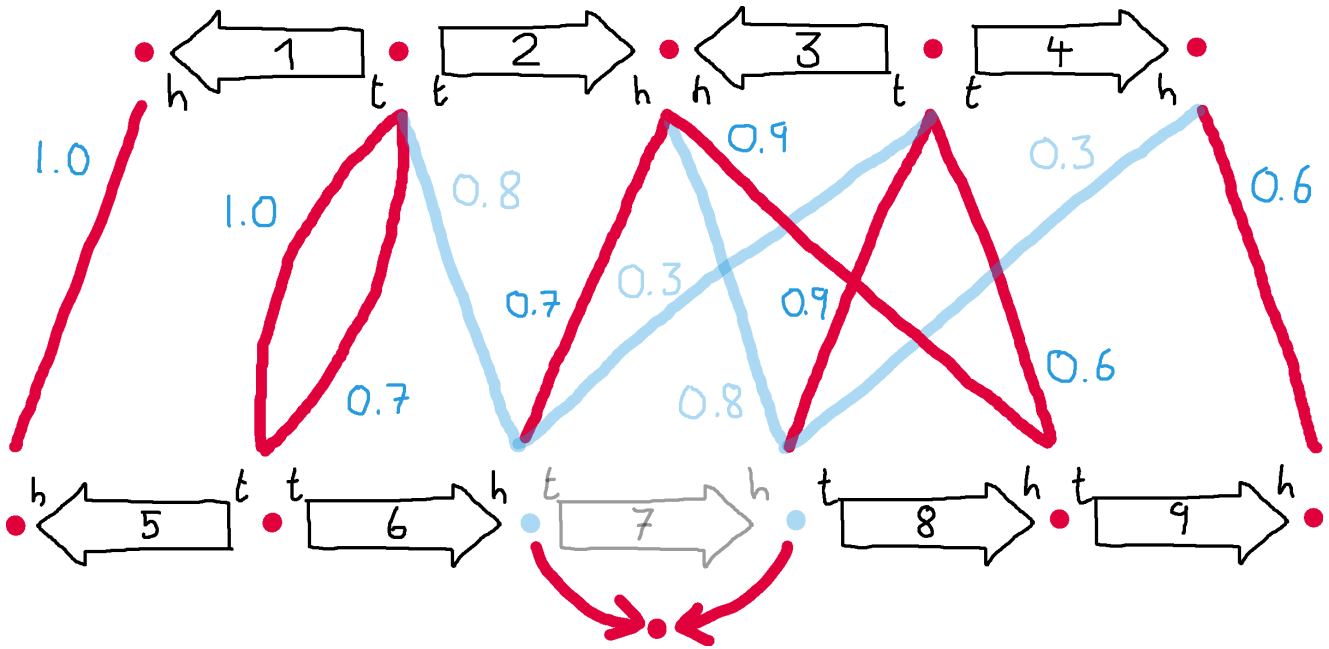
DCJ – Double Cut and Join

From the gene similarity graph to the **weighted adjacency graph** (WAG):



DCJ – Double Cut and Join

From the gene similarity graph to the **weighted adjacency graph** (WAG):



Family-free Rearrangement Problem

Family-free Rearrangement Problem

Find matching $\mathcal{M}_{GH}$ that maximizes the following formula:

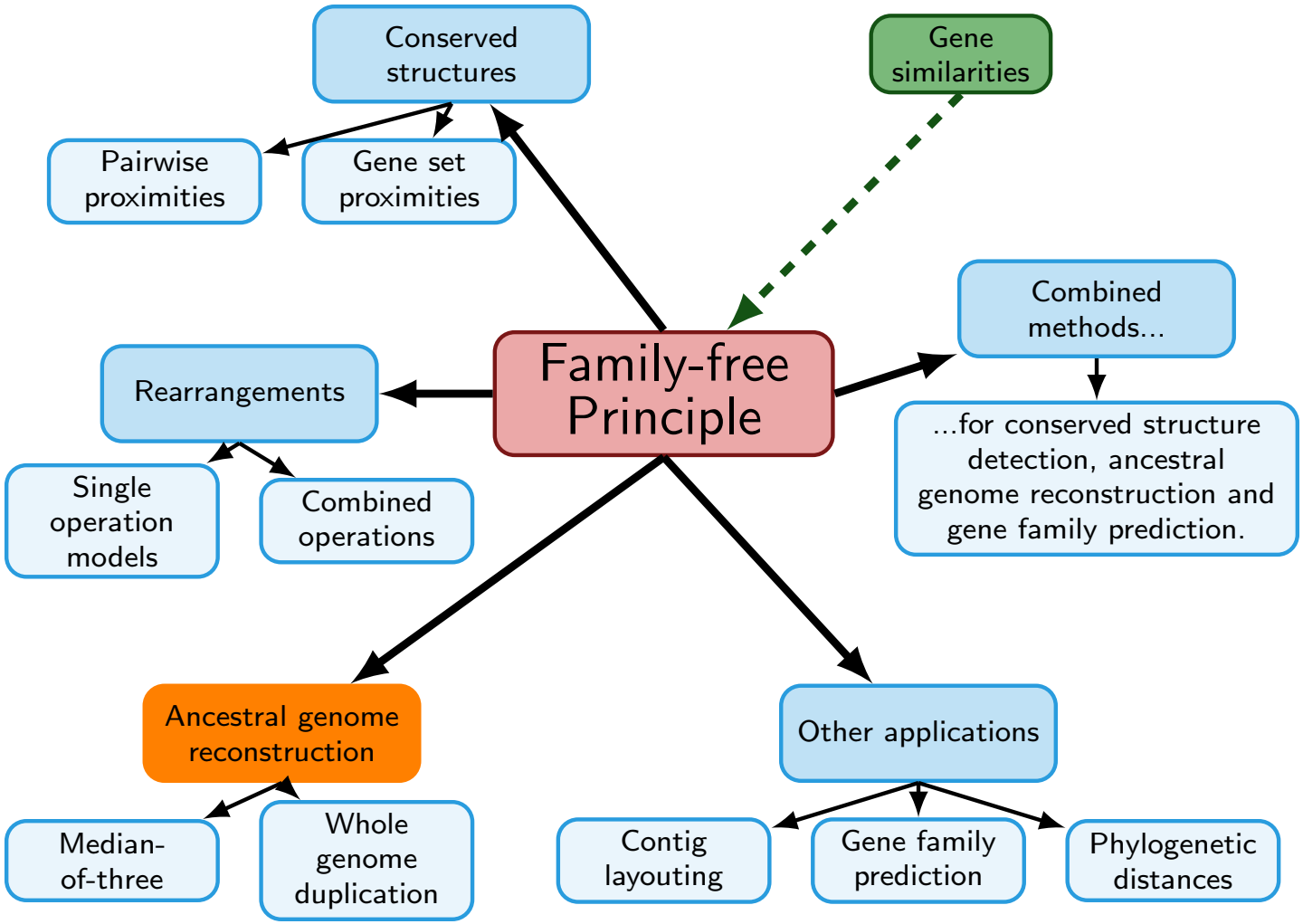
$$\mathcal{F}_{\alpha}^{DCJ}(\mathcal{M}_{GH}) = \alpha \cdot cyc(\mathcal{M}_{GH}) + (1-\alpha) \cdot edg(\mathcal{M}_{GH})$$

where

$$cyc(\mathcal{M}_{GH}) = \sum_{C \in \mathcal{C}(\mathcal{M}_{GH})} \left(\frac{1}{|C|} \sum_{e \in C} w(e) \right)$$

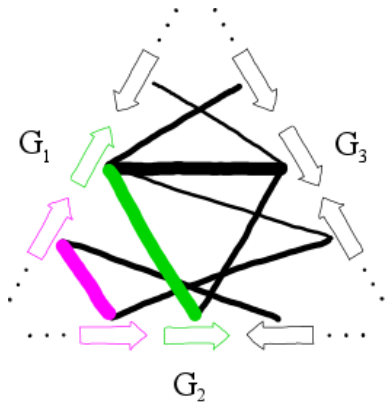
$\mathcal{C}(\mathcal{M}_{GH}) :=$ set of connected components in $WAG(\mathcal{M}_{GH})$

Ancestral genome reconstruction



Reconstruction of Ancestral Adjacencies

Emphasize adjacencies that are conserved in closely related genomes.



Phylogeny Aware Optimization Problem

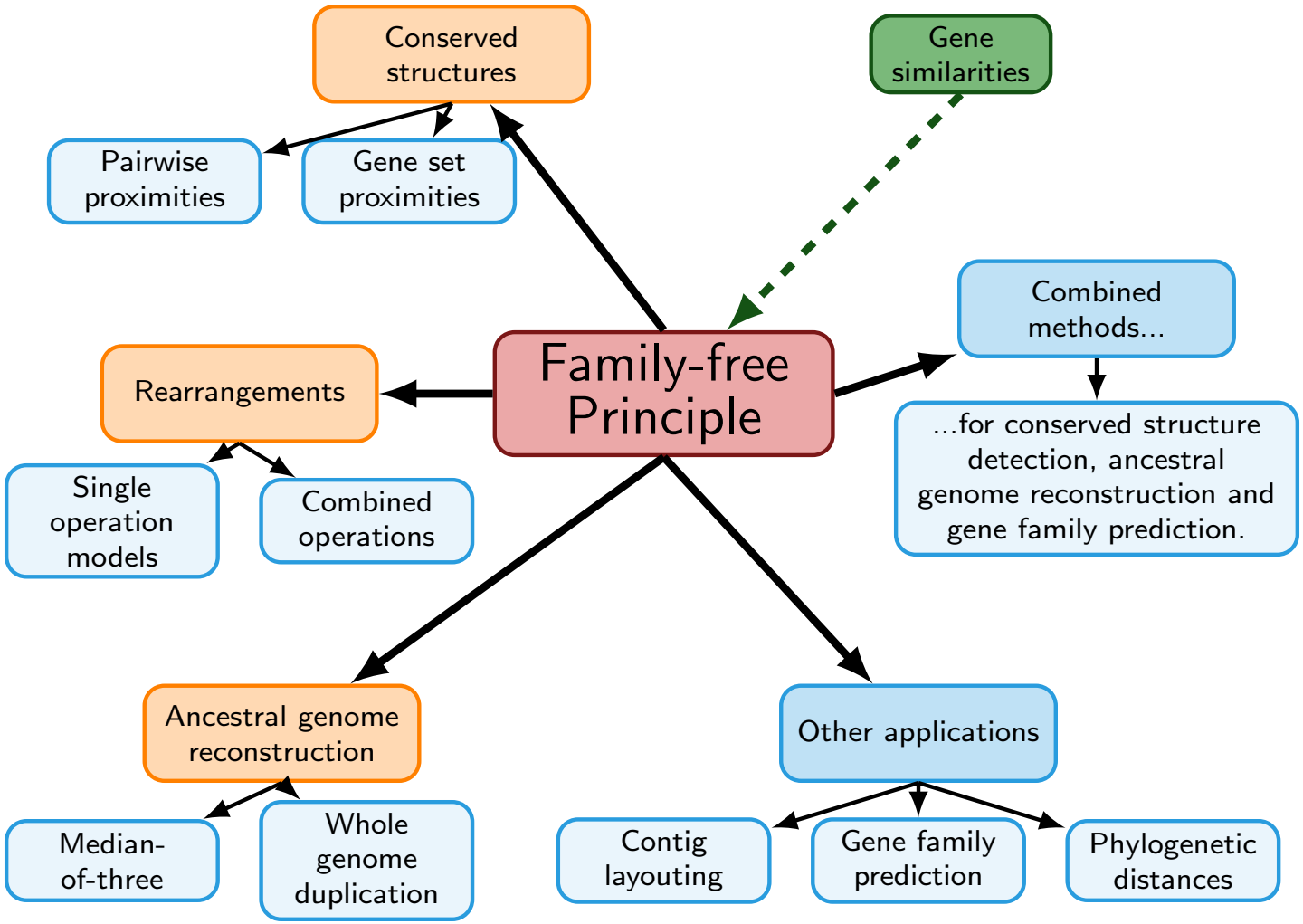
Given an additive distance matrix $D^{\mathcal{T}}$, find matching $\mathcal{M}$ that maximizes the following formula:

$$\mathcal{F}_{\alpha,\mathcal{T}}(\mathcal{M}) = \sum_{G,H}\left((D^{\mathcal{T}}_{max} - D^{\mathcal{T}}_{GH})\left(\alpha \cdot adj(\mathcal{M}_{GH}) + (1-\alpha)\cdot edg(\mathcal{M}_{GH})\right)\right)$$

where

$$D^{\mathcal{T}}_{max} = \max_{G,H}\{D^{\mathcal{T}}_{GH}\}$$

Conclusion and outlook



Thanks to:

Marília D. V. Braga

Cedric Chauve

Daniel Doerr

Katharina Jahn

Annelyse Thévenin

Roland Wittler

Thanks to:

Marília D. V. Braga

Cedric Chauve

Daniel Doerr

Katharina Jahn

Annelyse Thévenin

Roland Wittler

You!