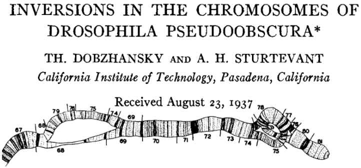
Quantifying Rearrangements in Pangenomes

Leonard Bohnenkämper

Bielefeld University

May 15th, 2025

Rearrangements



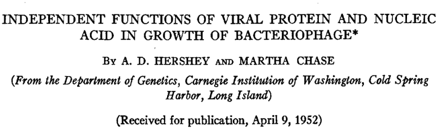
INVERSIONS IN THE CHROMOSOMES OF DROSOPHILA PSEUDOOBSCURA*

TH. DOBZHANSKY AND A. H. STURTEVANT

California Institute of Technology, Pasadena, California

Received August 23, 1937

79
78
75
74
69
68
67
68
69
70
71
72
73
74
75
78
77
76
80
81



INDEPENDENT FUNCTIONS OF VIRAL PROTEIN AND NUCLEIC ACID IN GROWTH OF BACTERIOPHAGE*

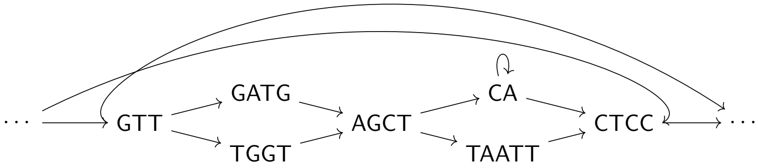
BY A. D. HERSHEY AND MARTHA CHASE

(*From the Department of Genetics, Carnegie Institution of Washington, Cold Spring Harbor, Long Island*)

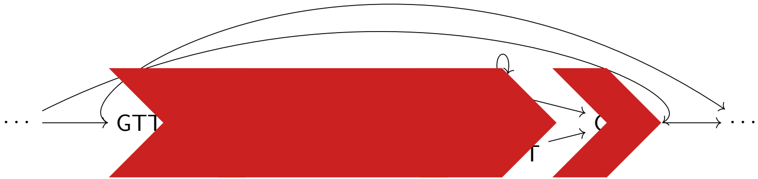
(Received for publication, April 9, 1952)

- ∼30,000 inversions affect more bases of the human genome than ∼5,000,000 SNVs
- Pivotal role in:
  - Evolution
  - Disease
- Rearrangement Quantification for Structural Variants ≈ Alignment for SNVs

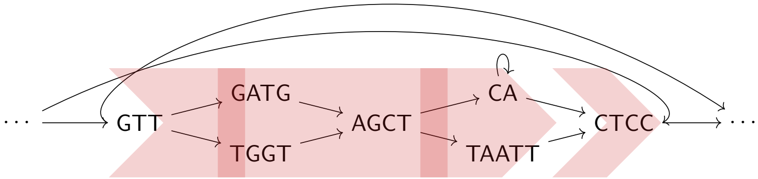
Abstracting from Local Variations



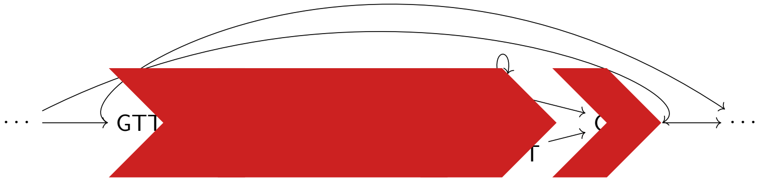
Abstracting from Local Variations



Abstracting from Local Variations



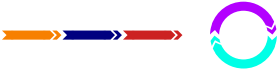
Abstracting from Local Variations



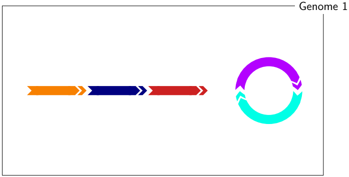
→ Collinear Block Detection.

Marker, Chromosome, Genome, Pangenome

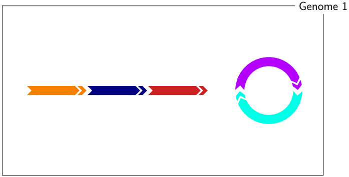
Marker, Chromosome, Genome, Pangenome

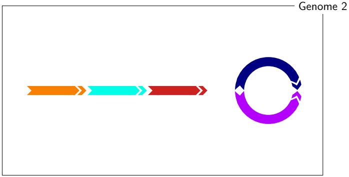


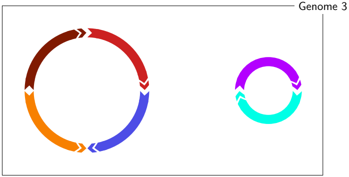
Marker, Chromosome, Genome, Pangenome

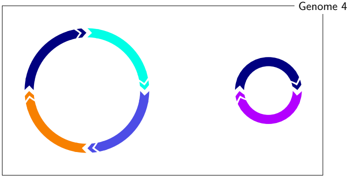


Marker, Chromosome, Genome, Pangenome

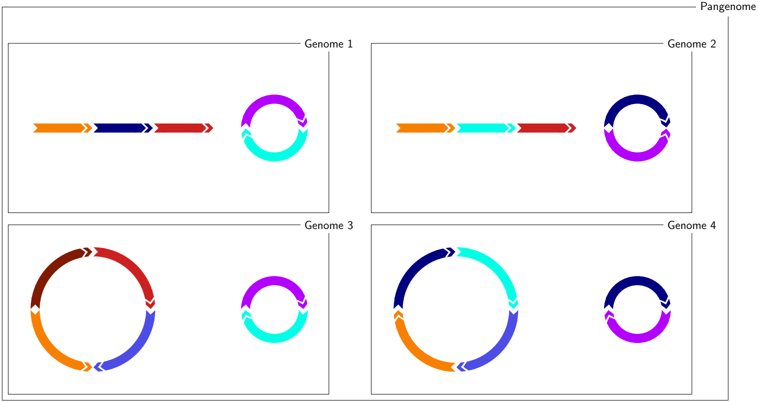




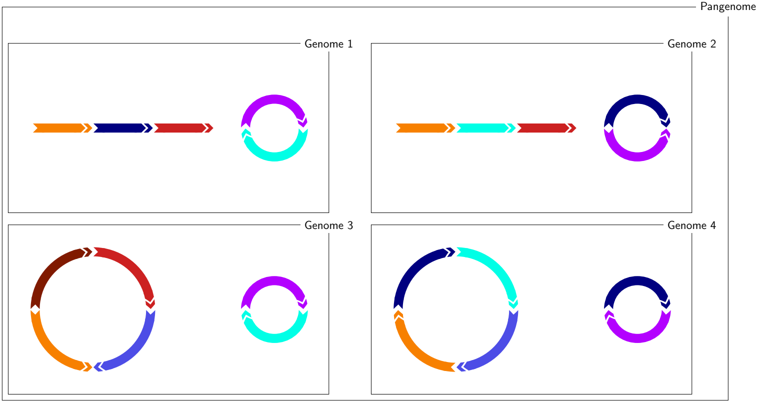




Marker, Chromosome, Genome, Pangenome



Marker, Chromosome, Genome, Pangenome



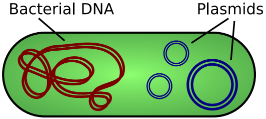
Classical models: Only one copy of each marker per genome.

SPP - Ancestral Reconstruction with a Given Tree

Joint work with Daria Frolova (, Jens Stoye, Daniel Doerr)

Background: Plasmids

- Extrachromosomal DNA
- Common in bacteria
- Mostly circular
- Some plasmids can be transferred between hosts (even cross-species)
- Often carry resistance or virulence genes



Plasmid image by en: User:Spaully on English wikipedia, CC BY-SA 2.5, via Wikimedia Commons

Application: Plasmids

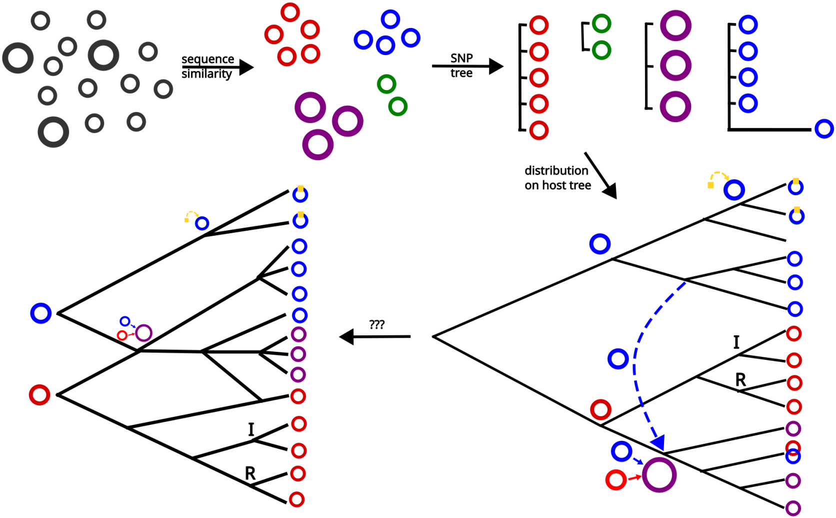
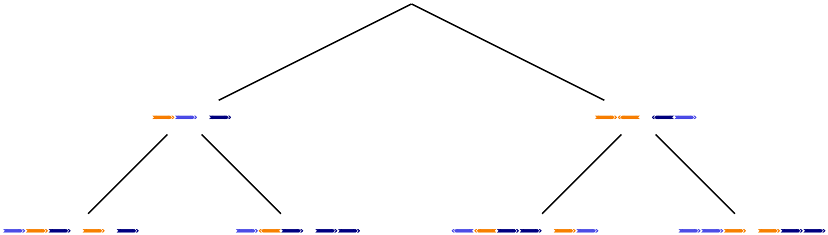
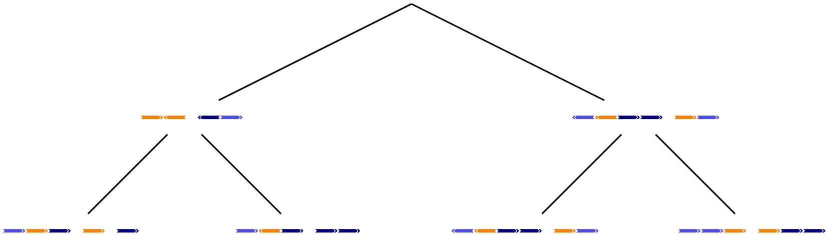


Image by courtesy of Daria Frolova

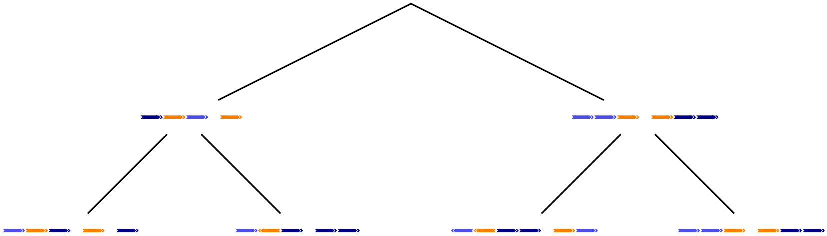
Small Parsimony for Natural Genomes



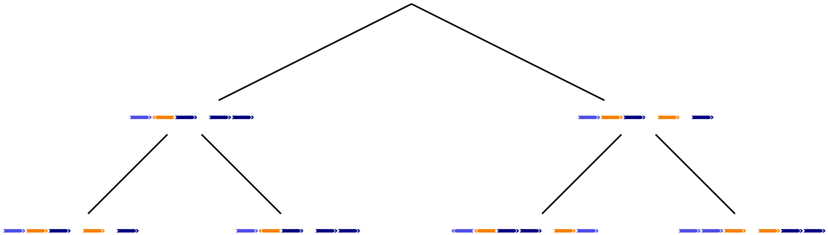
Small Parsimony for Natural Genomes



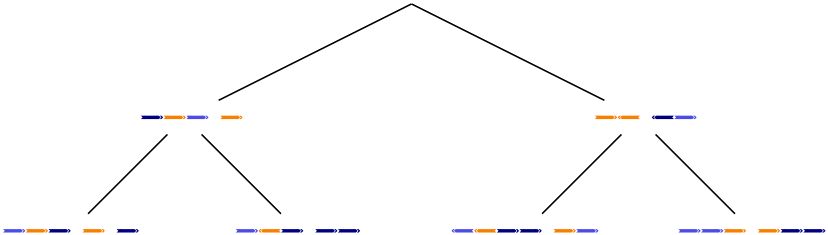
Small Parsimony for Natural Genomes



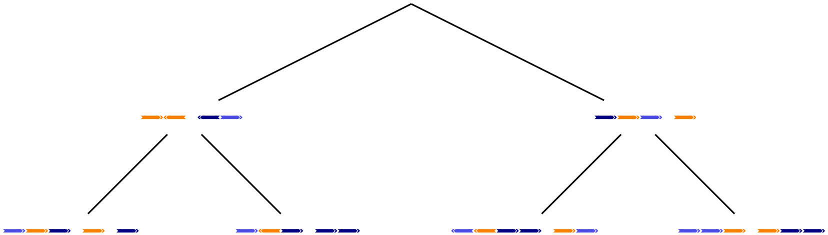
Small Parsimony for Natural Genomes



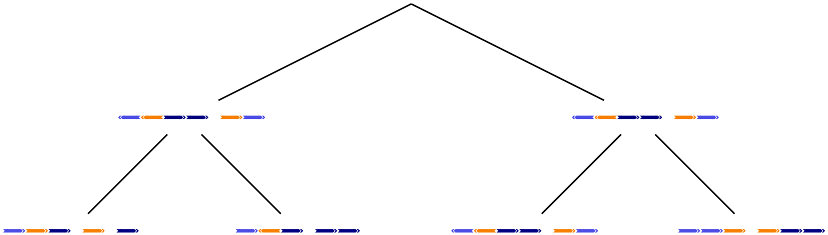
Small Parsimony for Natural Genomes



Small Parsimony for Natural Genomes

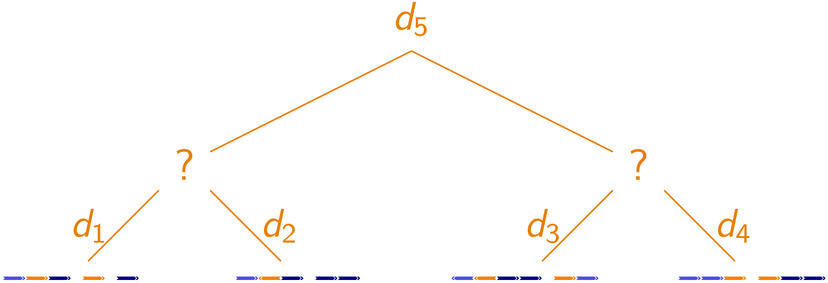


Small Parsimony for Natural Genomes

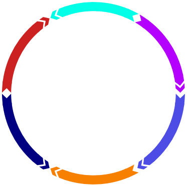


Small Parsimony for Natural Genomes

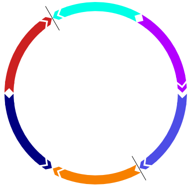
Minimize $d_1 + d_2 + d_3 + d_4 + d_5$



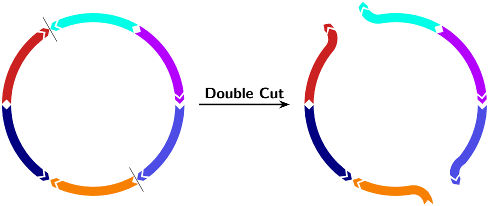
Model: Double-Cut-And-Join



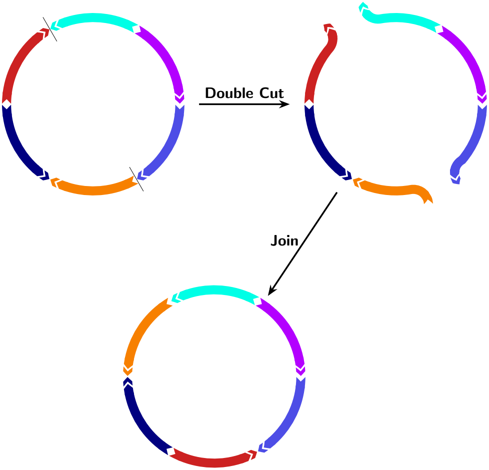
Model: Double-Cut-And-Join



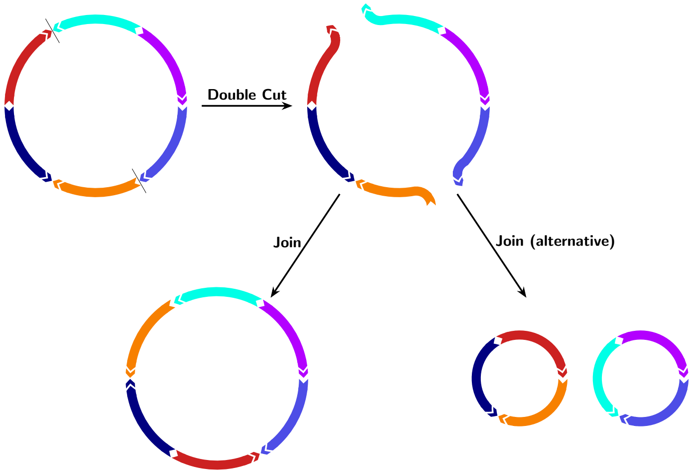
Model: Double-Cut-And-Join



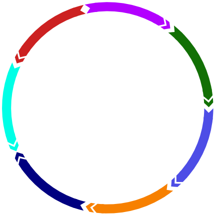
Model: Double-Cut-And-Join



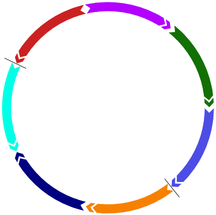
Model: Double-Cut-And-Join



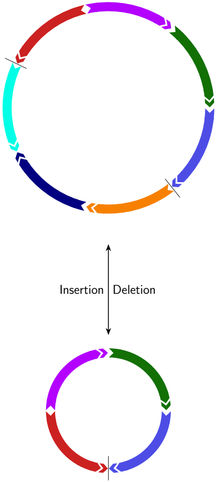
Insertions and Deletions



Insertions and Deletions

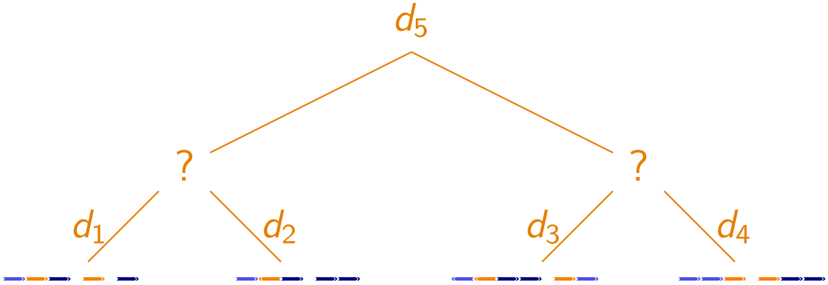


Insertions and Deletions



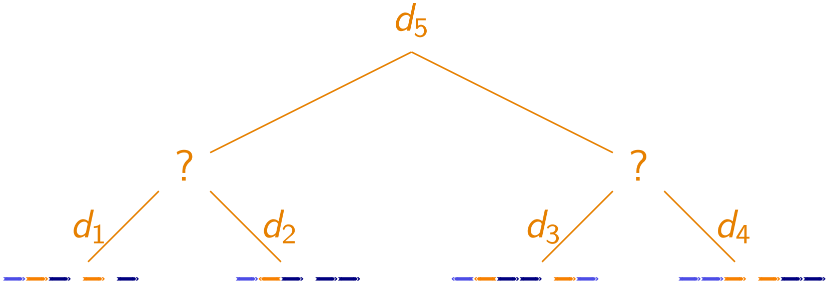
Small Parsimony for Natural Genomes

Minimize $d_1 + d_2 + d_3 + d_4 + d_5$



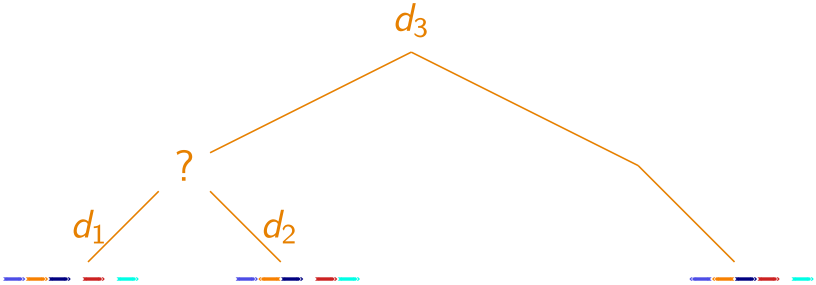
Small Parsimony for Natural Genomes

Minimize $d_1 + d_2 + d_3 + d_4 + d_5$

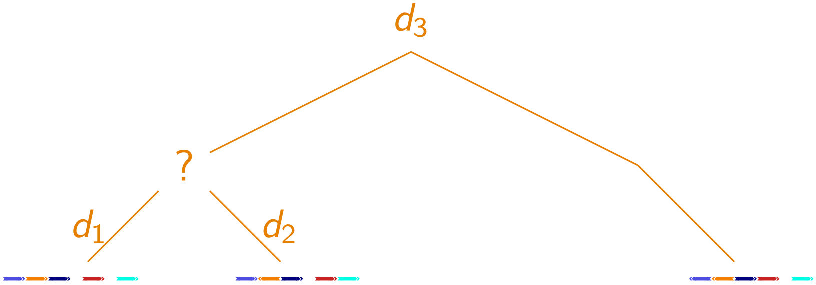


d_1, d_2, d_3, d_4, d_5: (Minimum number of) DCJ and indel operations between the genomes.

Small Parsimony for Natural Genomes

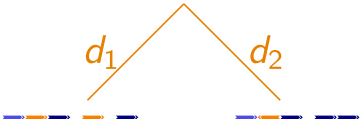


Small Parsimony for Natural Genomes

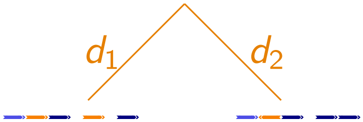


→ NP Hard

Small Parsimony for Natural Genomes

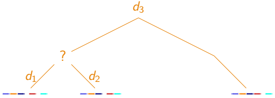


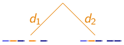
Small Parsimony for Natural Genomes



$\rightarrow$ NP Hard

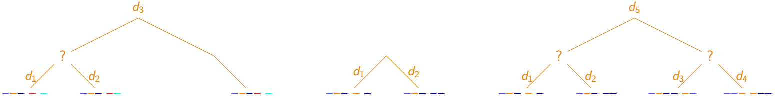
Small Parsimony for Natural Genomes





→ NP Hard → NP Hard

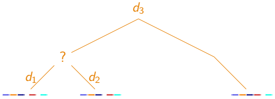
Small Parsimony for Natural Genomes



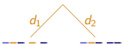
$\rightarrow$ NP Hard

$\rightarrow$ NP Hard

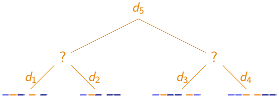
Small Parsimony for Natural Genomes



→ NP Hard

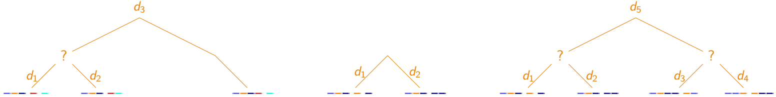


→ NP Hard



→ NP Hard

Small Parsimony for Natural Genomes



→ NP Hard → NP Hard → NP Hard

→ Solution via ILP

Algorithm 1 Capping-free Small Parsimony

Objective

`Minimize`

$$\sum_{E \in E(T)} (\alpha\, \mathtt{f}_E + (\alpha - 1)\, \mathtt{w}_E)$$

Global level

For each genome $\mathbb{A} = (\mathcal{E} \cup \mathcal{T}, \mathcal{M} \cup \mathcal{A})$ of phylogeny T:

(C.01)	$\mathtt{g}_u = \mathtt{g}_v$	with $(u,v) \in \mathcal{M}$
(C.02)	$\sum_{m \in F} \mathtt{g}_{m^h} \geq L_F^{\mathbb{A}}$ $\sum_{m \in F} \mathtt{g}_{m^h} \leq H_F^{\mathbb{A}}$	for each family F
(C.03)	$\sum_{uv \in \mathcal{A}} \mathtt{a}_{uv} = \mathtt{g}_v$	$\forall v \in \mathcal{E} \cup \mathcal{T}$

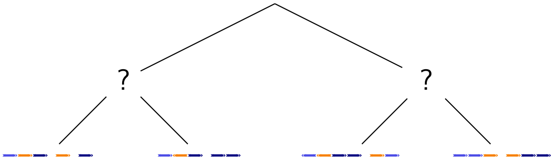
Local level

For each edge $E = (\mathbb{A}, \mathbb{B}) \in E(T)$ with $\mathcal{CFMRD}(\mathbb{A}, \mathbb{B}) = (\mathcal{E} \cup \mathcal{T}, E_{\text{adj}}$ $E_{\text{ext}} \cup E_{\text{self}})$:

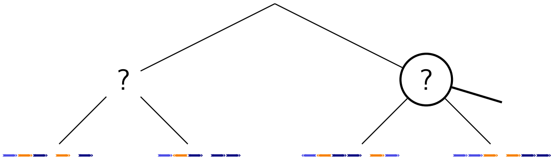
(C.04)	$\mathtt{w}_E = \sum_{e \in E_{\text{adj}}} \mathbf{w}(e) x_e$
(C.05)	$\mathtt{f}_E = \mathtt{n}_E - \mathtt{c}_E + \mathtt{q}_E + \mathtt{s}_E$
(C.06)	$\mathtt{n}_E = \frac{1}{2} \sum_{e \in E_{\text{ext}}} \mathtt{x}_e$
(C.07)	$\mathtt{c}_E = \sum_{v \in \mathcal{E}} \mathtt{r}_v^c$
(C.08)	$2\mathtt{q}_E \geq \mathtt{p}_E^{ab} + \mathtt{p}_E^{\max a} + \mathtt{p}_E^{\max b} - \mathtt{p}_E^{AB}$

...

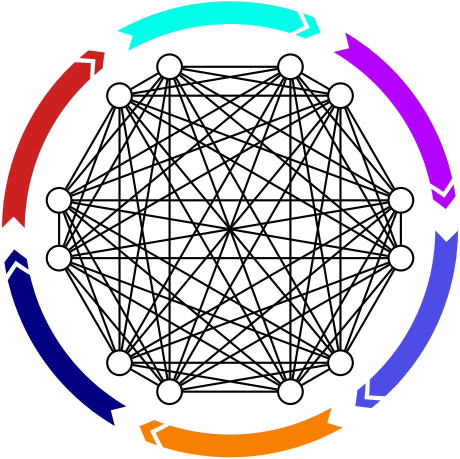
Vast Solution Space for Ancestral Adjacencies



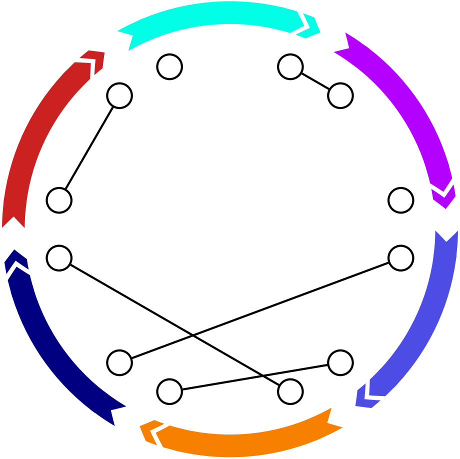
Vast Solution Space for Ancestral Adjacencies



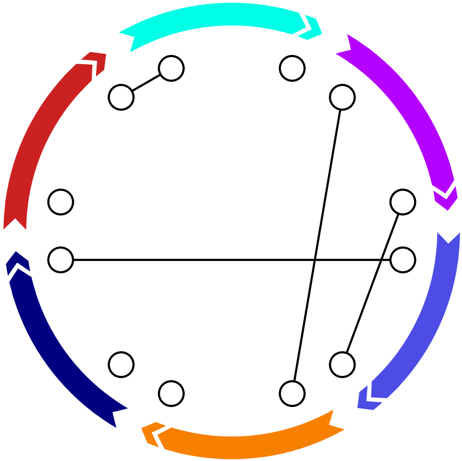
Vast Solution Space for Ancestral Adjacencies



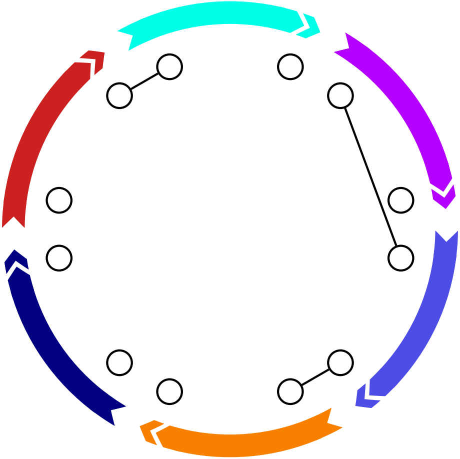
Vast Solution Space for Ancestral Adjacencies



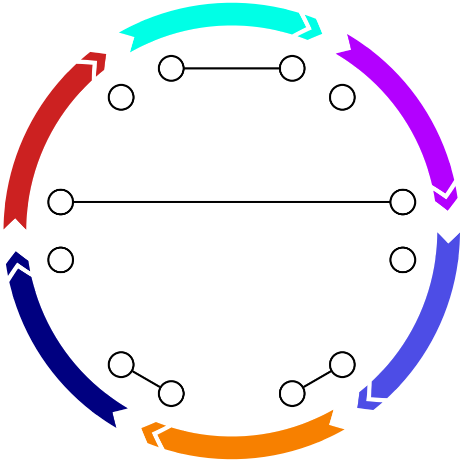
Vast Solution Space for Ancestral Adjacencies



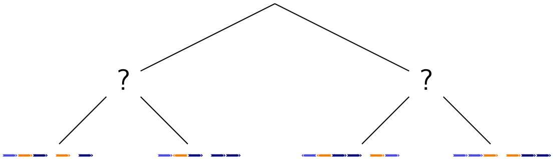
Vast Solution Space for Ancestral Adjacencies



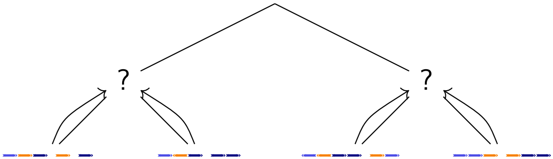
Vast Solution Space for Ancestral Adjacencies



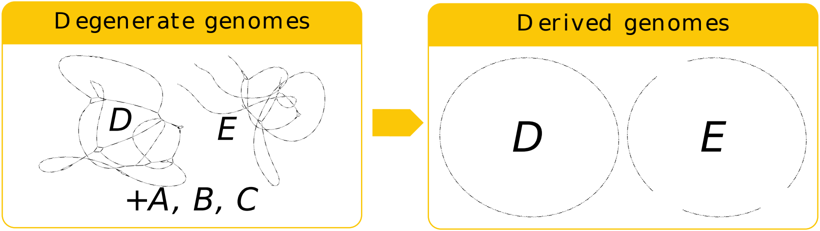
Heuristic Solution: Filter Adjacencies Based on Leafs



Heuristic Solution: Filter Adjacencies Based on Leafs



Degenerate Genomes



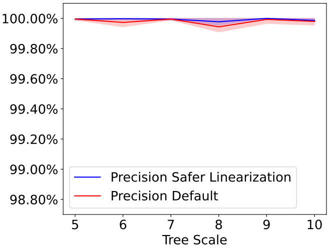
Degenerate genomes

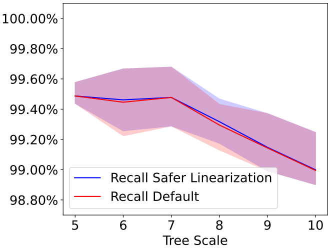
Derived genomes

(Slightly) Adapted from (Doerr, Chauve, 2021).

Degenerate Genomes - the Good

On simulated data (ZOMBI+pre-filtering with DeCoSTAR), *if the ground truth adjacencies are represented in the degenerate genome*:





Degenerate Genomes - the Bad

- Good pre-filtering (DeCoSTAR) requires gene trees.

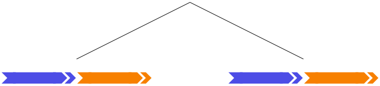
- Is not exact.

Our idea: Filter and Solver Tricks

Our idea: Filter and Solver Tricks

- Filter out adjacencies that are provably not needed.
  → Exploit features of the data:

Our idea: Filter and Solver Tricks

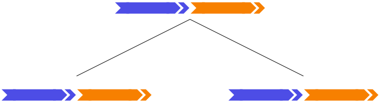


- Filter out adjacencies that are provably not needed.

  → Exploit features of the data:

  - Conserved adjacencies at the leaves

Our idea: Filter and Solver Tricks

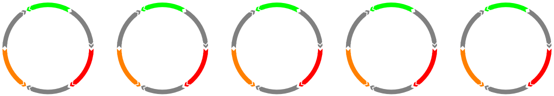


- Filter out adjacencies that are provably not needed.

  → Exploit features of the data:

  - Conserved adjacencies at the leaves

Our idea: Filter and Solver Tricks

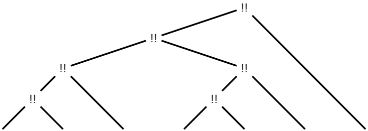


- Filter out adjacencies that are provably not needed.

  → Exploit features of the data:

  - Conserved adjacencies at the leaves
  - Conserved "Core" structure

Our idea: Filter and Solver Tricks



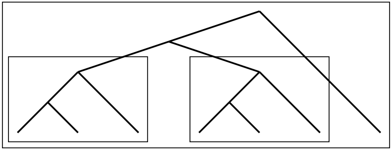
- Filter out adjacencies that are provably not needed.

  → Exploit features of the data:

  - Conserved adjacencies at the leaves
  - Conserved “Core” structure

- Solver Tricks:
  - Good solutions are easily guessed → “warm start”

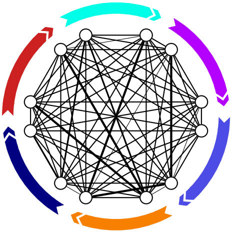
Our idea: Filter and Solver Tricks



- Filter out adjacencies that are provably not needed.

  → Exploit features of the data:
  - Conserved adjacencies at the leaves
  - Conserved "Core" structure
- Solver Tricks:
  - Good solutions are easily guessed → "warm start"
  - Solve progressively more complex versions of the problem:
    - Subtrees

Our idea: Filter and Solver Tricks



- Filter out adjacencies that are provably not needed.

  → Exploit features of the data:

  - Conserved adjacencies at the leaves
  - Conserved "Core" structure
- Solver Tricks:
  - Good solutions are easily guessed → "warm start"
  - Solve progressively more complex versions of the problem:
    - Subtrees
    - Weighted adjacencies (strong weights → no weights)

Our idea: Filter and Solver Tricks

- Filter out adjacencies that are provably not needed.

  → Exploit features of the data:
  - Conserved adjacencies at the leaves
  - Conserved “Core” structure
- Solver Tricks:
  - Good solutions are easily guessed → “warm start”
  - Solve progressively more complex versions of the problem:
    - Subtrees
    - Weighted adjacencies (strong weights → no weights)

Our idea: Filter and Solver Tricks

- Filter out adjacencies that are provably not needed.

  → Exploit features of the data:
  - Conserved adjacencies at the leaves ✓
  - Conserved "Core" structure
- Solver Tricks:
  - Good solutions are easily guessed → "warm start"
  - Solve progressively more complex versions of the problem:
    - Subtrees
    - Weighted adjacencies (strong weights → no weights) ✓

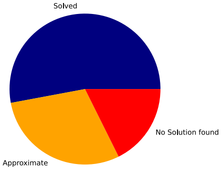
Extremely Preliminary Results

gurobi 12 on 1 thread with 1 hour time limit on all lineages (17) with 20 leaves or fewer

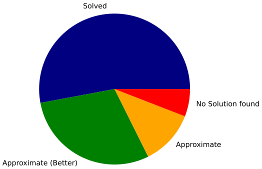
Extremely Preliminary Results

gurobi 12 on 1 thread with 1 hour time limit on all lineages (17) with 20 leaves or fewer

Unoptimized

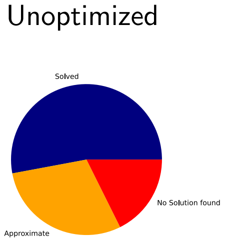


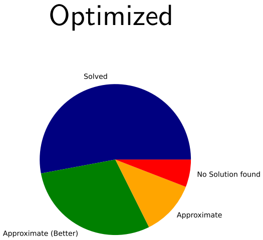
Optimized



Extremely Preliminary Results

gurobi 12 on 1 thread with 1 hour time limit on all lineages (17) with 20 leaves or fewer





- Main problem seems to be closing the gap.

SPP - Long Term Perspectives

- Transfer bounds from less complex (sub-) problems

SPP - Long Term Perspectives

- Transfer bounds from less complex (sub-) problems
- ILP solving $\overset{?}{\rightleftarrows}$ SAT solving

SPP - Long Term Perspectives

- Transfer bounds from less complex (sub-) problems
- ILP solving $\overset{?}{\rightleftarrows}$ SAT solving
- SPP-variant that can model other types of events (horizontal transfers, chromosome duplication,...)

SPP - Long Term Perspectives

- Transfer bounds from less complex (sub-) problems
- ILP solving $\overset{?}{\rightleftarrows}$ SAT solving
- SPP-variant that can model other types of events (horizontal transfers, chromosome duplication,...)
- Large Parsimony problem (no given tree)

CARP - A Specialized Problem for Pangenomes

Joint work with Jens Stoye

Quantifying Rearrangement Complexity in Pangenomes - Intuition

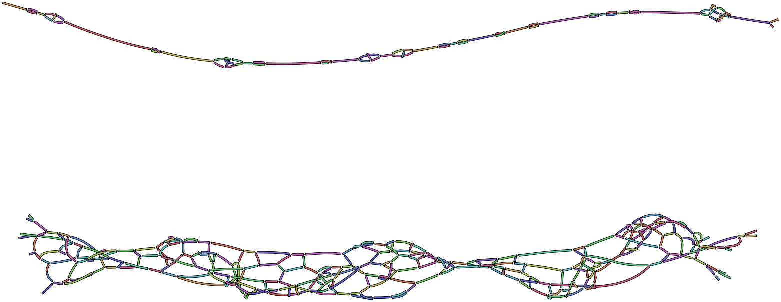
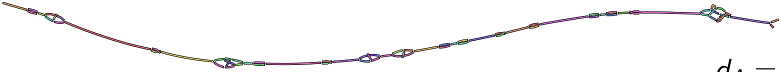
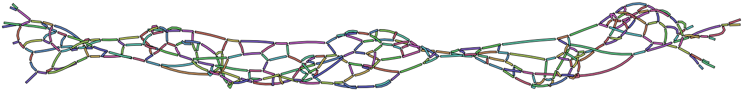


Image by courtesy of Luca Parmigiani

Quantifying Rearrangement Complexity in Pangenomes - Intuition



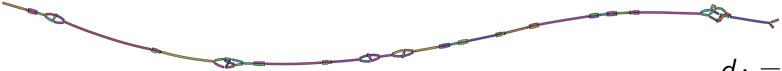
$d_A = 14$



$d_B = 107$

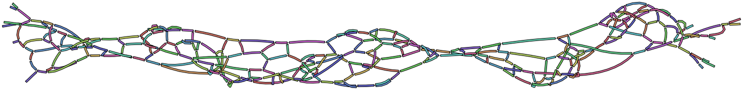
Image by courtesy of Luca Parmigiani

Quantifying Rearrangement Complexity in Pangenomes - Intuition



$d_A = ??$

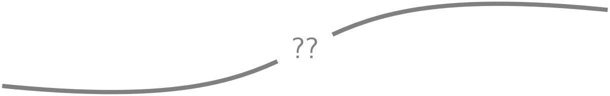
$d_A > d_B$



$d_B = ??$

Image by courtesy of Luca Parmigiani

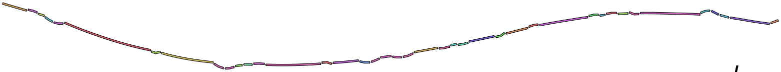
Quantifying Rearrangement Complexity in Pangenomes - Strategy



$d_X = 0$

Quantifying Rearrangement Complexity in Pangenomes - Strategy

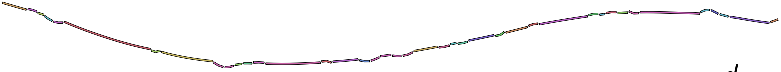
~~Boring~~ **Simple Pangenome**



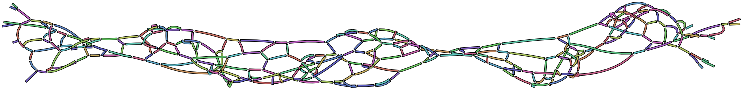
$d_X = 0$

Quantifying Rearrangement Complexity in Pangenomes - Strategy

~~Boring~~ **Simple Pangenome**

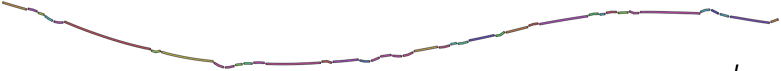


$d_X = 0$



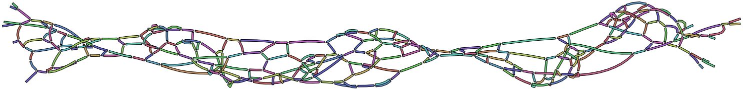
Quantifying Rearrangement Complexity in Pangenomes - Strategy

~~Boring~~ **Simple Pangenome**



$d_X = 0$

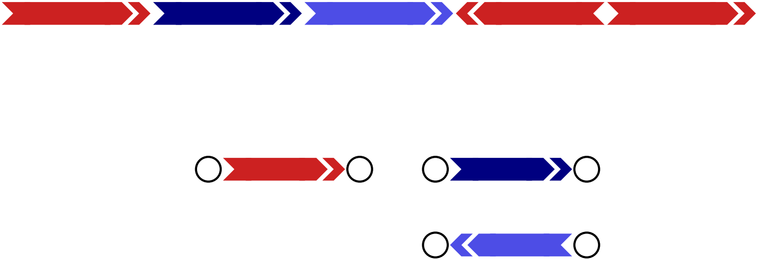
↑ Transform by operations T



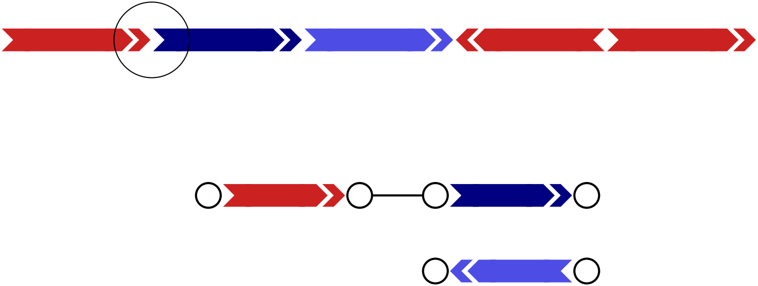
$d_B = |T|$

Multi-Breakpoint Graph (MBPG)

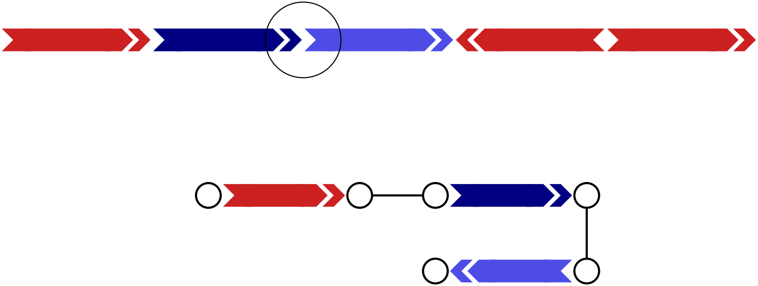
Multi-Breakpoint Graph (MBPG)



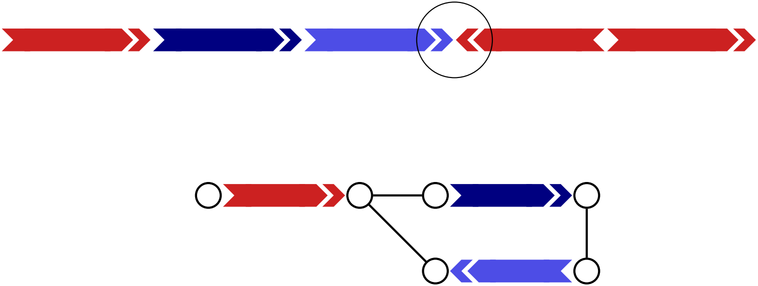
Multi-Breakpoint Graph (MBPG)



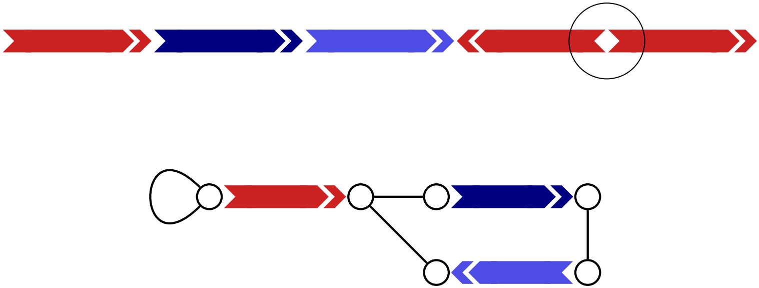
Multi-Breakpoint Graph (MBPG)



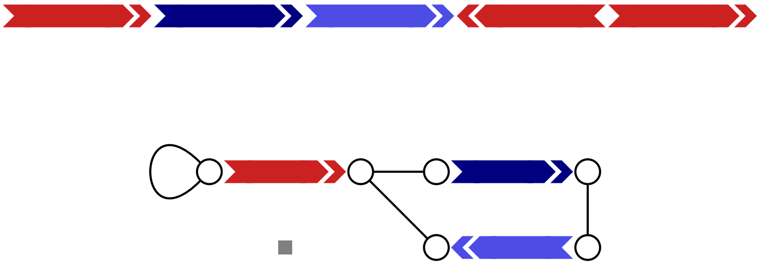
Multi-Breakpoint Graph (MBPG)



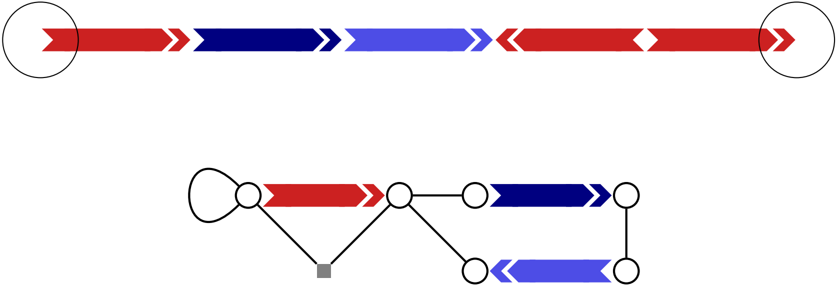
Multi-Breakpoint Graph (MBPG)



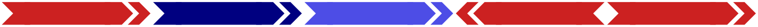
Multi-Breakpoint Graph (MBPG)

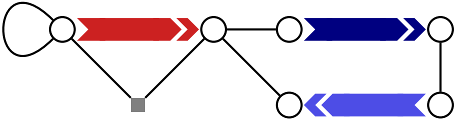


Multi-Breakpoint Graph (MBPG)



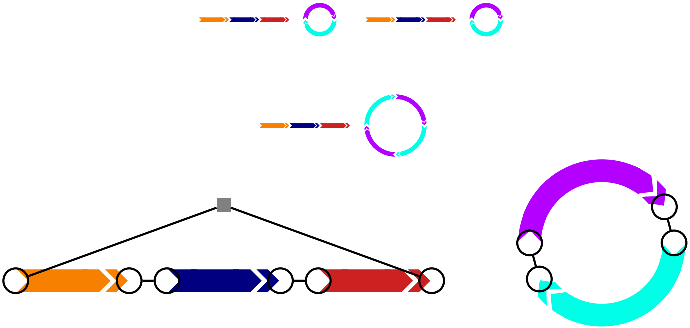
Multi-Breakpoint Graph (MBPG)





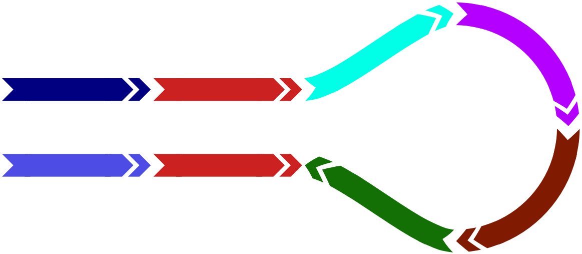
- GFA format (w/o sequence) can be represented as an MBPG
- Note: We still assume local variants have been filtered

MBPG of a Simple Pangenome

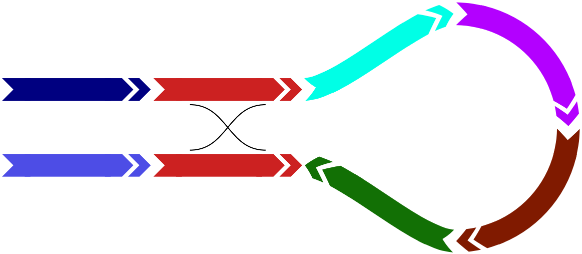


- Simple pangenome $\iff$ adjacencies are a perfect matching

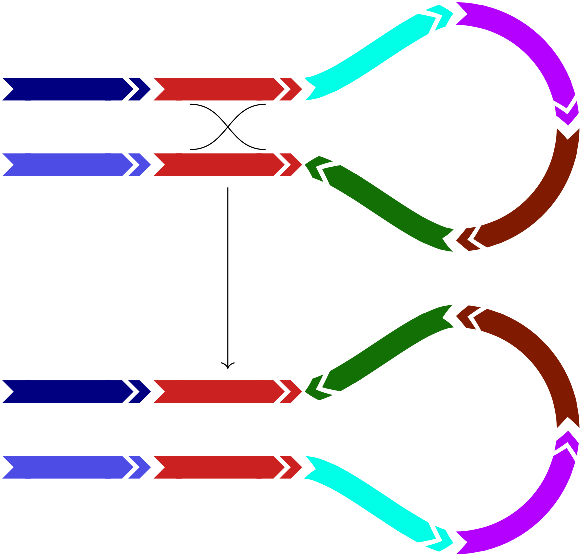
Rearrangements and Homologous Recombinations



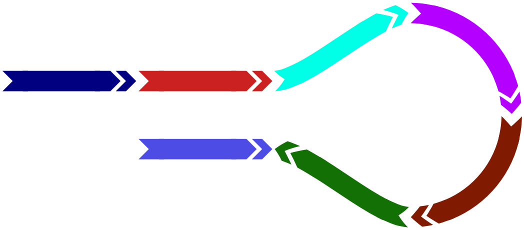
Rearrangements and Homologous Recombinations



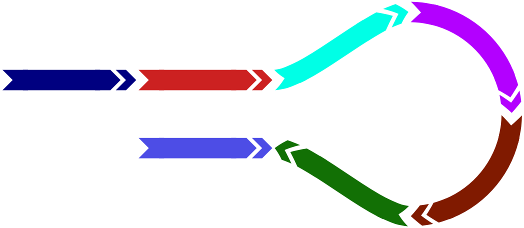
Rearrangements and Homologous Recombinations



Rearrangements and Homologous Recombinations

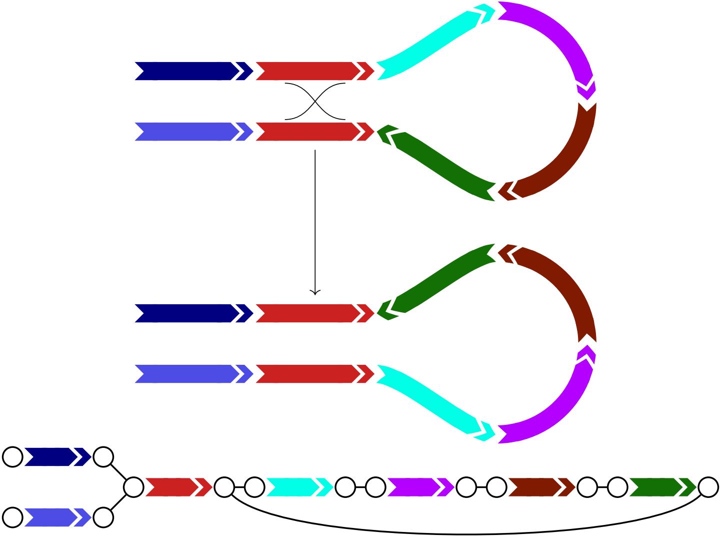


Rearrangements and Homologous Recombinations



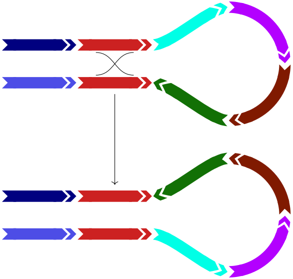
- No rearrangements via Homologous Recombination.

MBPG and Homologous Recombinations

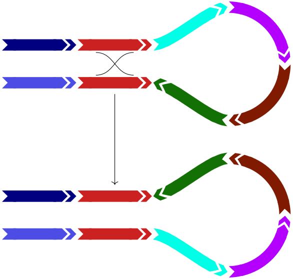


- Homologous Recombinations don't change the graph!

(Non)-Adjacency Modifying Operations

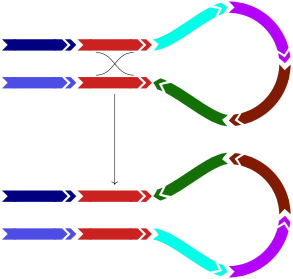


(Non)-Adjacency Modifying Operations



- No change in adjacencies!

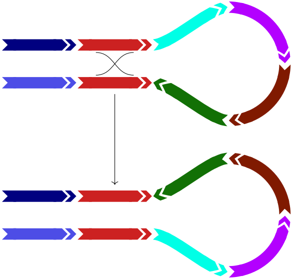
(Non)-Adjacency Modifying Operations



- No change in adjacencies!

  → Non-Adjacency Modifying Operations (Namos)

(Non)-Adjacency Modifying Operations

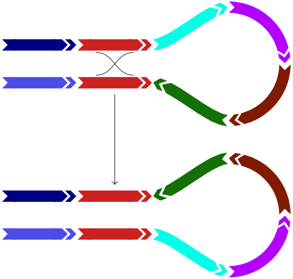


- No change in adjacencies!

  → Non-Adjacency Modifying Operations (Namos)

  ↔ Adjacency Modifying Operations (Amos)

(Non)-Adjacency Modifying Operations



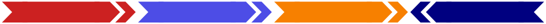
- No change in adjacencies!

  → Non-Adjacency Modifying Operations (Namos)

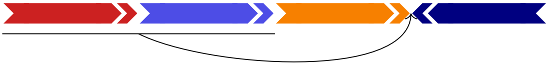
  ↔ Adjacency Modifying Operations (Amos)

Amos can introduce structural complexity!

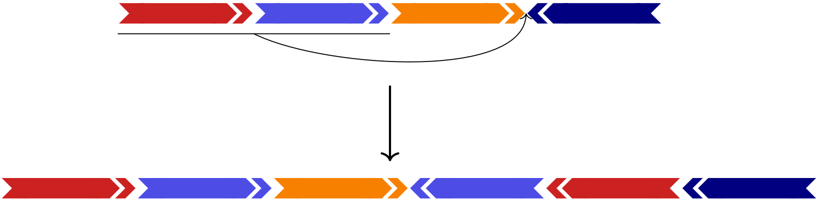
Amos - Example: Segmental Duplication



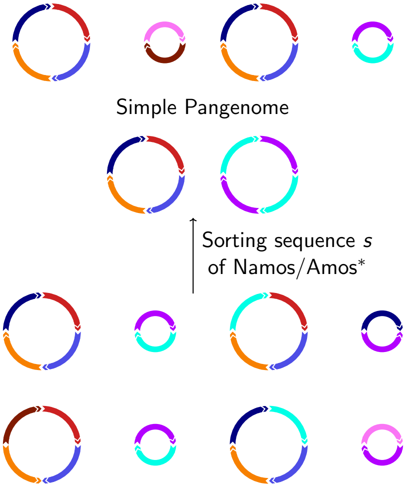
Amos - Example: Segmental Duplication



Amos - Example: Segmental Duplication



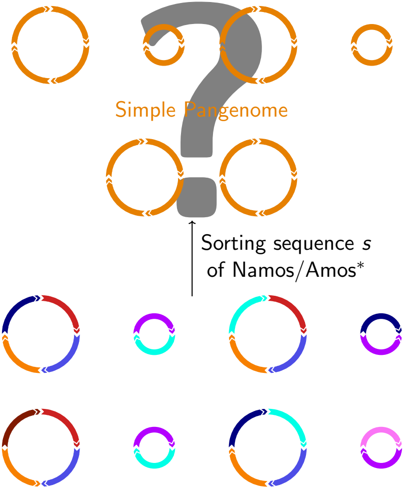
General Ancestral Reconstruction Problem (GARP)



Simple Pangenome

Sorting sequence *s* of Namos/Amos*

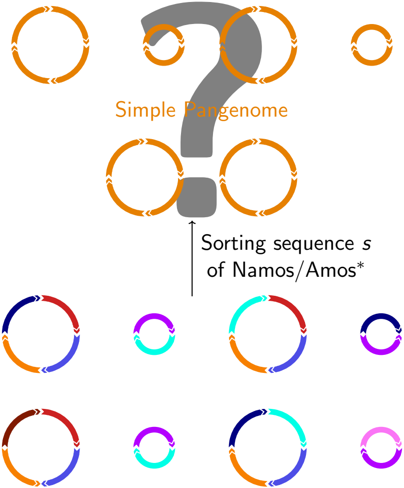
General Ancestral Reconstruction Problem (GARP)



Simple Pangenome

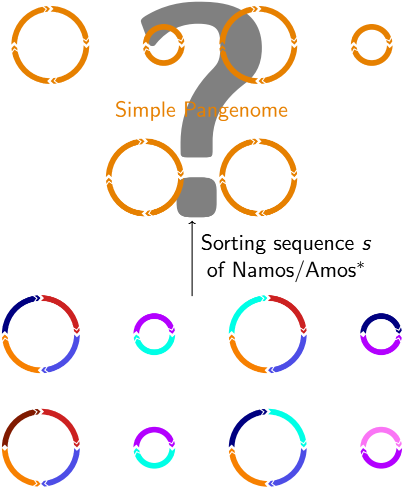
Sorting sequence *s* of Namos/Amos*

General Ancestral Reconstruction Problem (GARP)



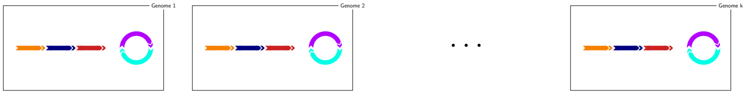
- #Amos in *s*: measure of the Structural Complexity.

General Ancestral Reconstruction Problem (GARP)

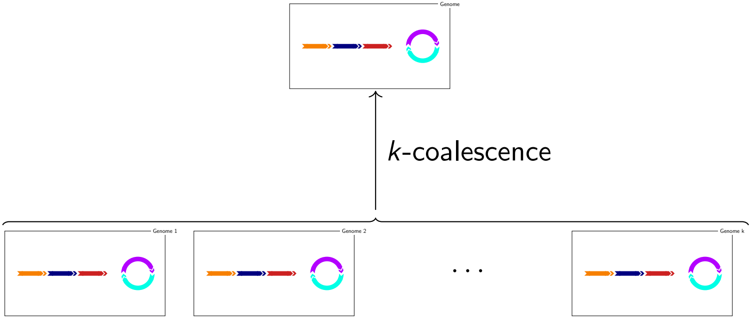


- #Amos in *s*: measure of the Structural Complexity.
- * with minimal #Amos

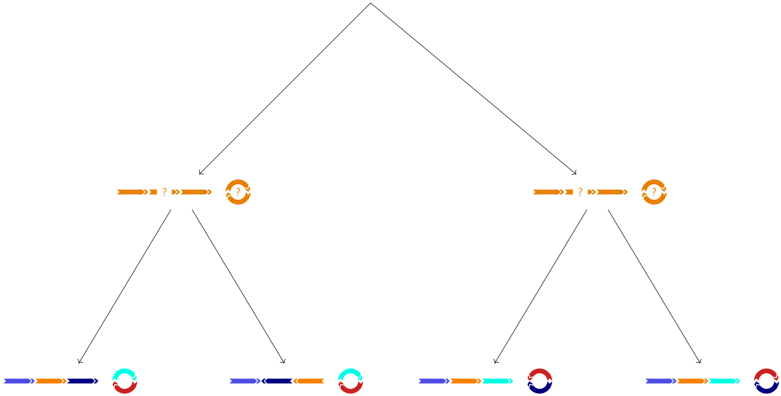
k-Coalescence – an important Namo



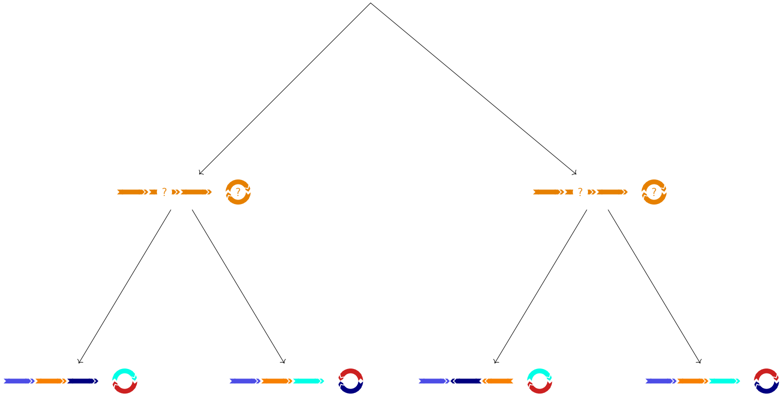
k-Coalescence – an important Namo



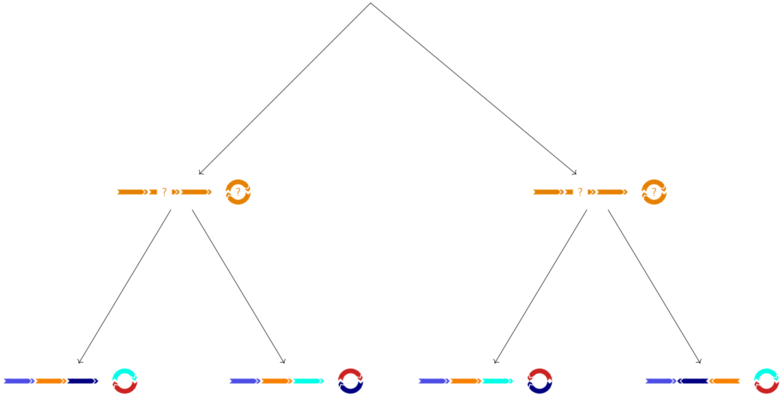
Large Parsimony Problem and GARP



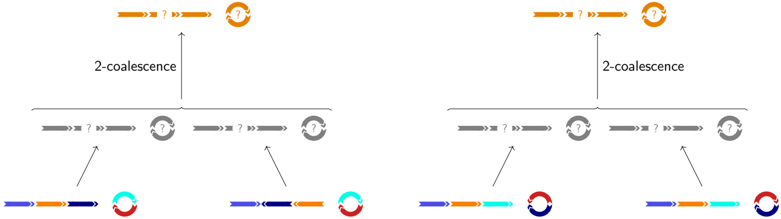
Large Parsimony Problem and GARP



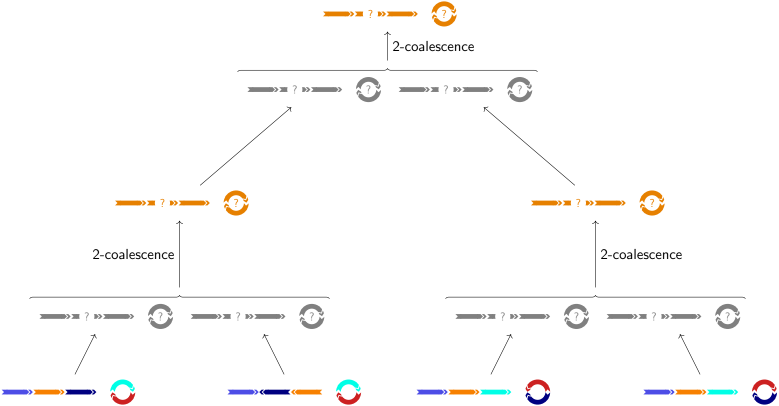
Large Parsimony Problem and GARP



Large Parsimony Problem and GARP



Large Parsimony Problem and GARP



GARP and Classical Problem Formulations

Problem name	Given genomes	Classical Formulation	Namos (N)
Distance problem	Genomes X, Y on the same marker set (single copy)	Find the minimum number of operations in M to transform X to Y.	– (2-coalescence)
Median problem	Genomes $X_1, \ldots, X_k$ on the same marker set (single copy)	Find a genome Y that minimizes the total distance to $X_1, \ldots, X_k$.	– (k-coalescence)
Large parsimony [illegible]	Genomes $X_1, \ldots, X_k$ on the [illegible] copy)	Find a binary tree with leaves $X_1, \ldots, X_k$ [illegible] minimal total distance between the genomes at each edge.	2-coalescence

- Very limited interaction between genomes (mostly coalescence, sometimes chromosome (de-)duplication)

GARP and Classical Problem Formulations

Problem name	Given genomes	Classical Formulation	Namos (N)
Distance problem	Genomes X, Y on the same marker set (single copy)	Find the minimum number of operations in M to transform X to Y.	– (2-coalescence)
Median problem	Genomes $X_1, \dots, X_k$ on the same marker set (single copy)	Find a genome Y that minimizes the total distance to $X_1, \dots, X_k$.	– (k-coalescence)
Large parsimony	Genomes $X_1, \dots, X_k$ on the [illegible] copy)	Find a binary tree with leaves $X_1, \dots, X_k$ [illegible] minimal total distance between the genomes at each edge.	2-coalescence

- Very limited interaction between genomes (mostly coalescence, sometimes chromosome (de-)duplication)
- Pangenomes → phylogenetic closeness → more interaction/horizontal effects

GARP and Classical Problem Formulations

Problem name	Given genomes	Classical Formulation	Namos (N)
Distance problem	Genomes X, Y on the same marker set (single copy)	Find the minimum number of operations in M to transform X to Y.	– (2-coalescence)
Median problem	Genomes $X_1, \dots, X_k$ on the same marker set (single copy)	Find a genome Y that minimizes the total distance to $X_1, \dots, X_k$.	– (k-coalescence)
Large parsimony problem	Genomes $X_1, \dots, X_k$ on the same marker set (single copy)	Find a binary tree with leaves $X_1, \dots, X_k$ and genomes $X_{k+1}, \dots, X_{2k-1}$ as inner nodes with minimal total distance between the genomes at each edge.	2-coalescence

- Very limited interaction between genomes (mostly coalescence, sometimes chromosome (de-)duplication)
- Pangenomes → phylogenetic closeness → more interaction/horizontal effects

→ Classical Problem Formulations would overestimate structural complexity for pangenomes (missing Namos substituted with Amos)

GARP and Classical Problem Formulations

Problem name	Given genomes	Classical Formulation	Namos (N)
Distance problem	Genomes X, Y on the same marker set (single copy)	Find the minimum number of operations in M to transform X to Y.	– (2-coalescence)
Median problem	Genomes $X_1, \dots, X_k$ on the same marker set (single copy)	Find a genome Y that minimizes the total distance to $X_1, \dots, X_k$.	– (k-coalescence)
Large parsimony	Genomes $X_1, \dots, X_k$ on the same marker set (single copy)	Find a binary tree with leaves $X_1, \dots, X_k$ [illegible] $_{k-1}$ as inner nodes [illegible] minimal total distance between the genomes at each edge.	2-coalescence

- ▶ Very limited interaction between genomes (mostly coalescence, sometimes chromosome (de-)duplication)
- ▶ Pangenomes → phylogenetic closeness → more interaction/horizontal effects

→ Classical Problem Formulations would overestimate structural complexity for pangenomes (missing Namos substituted with Amos)

→ For a lower bound, we need a “maximally powerful” set of Namos

Obervations from the MBPG

Observation

Given a Namo o and pangenomes $\mathbb{P}, \mathbb{P}'$, where $\mathbb{P} \xrightarrow{o} \mathbb{P}'$, the MBPGs of $\mathbb{P}$ and $\mathbb{P}'$ are identical.

Obervations from the MBPG

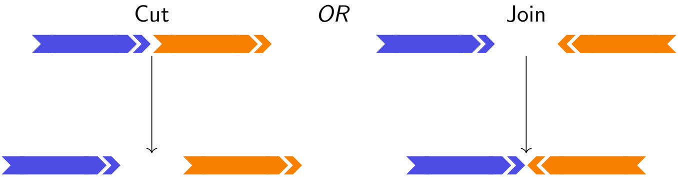
Observation

Given a Namo o and pangenomes $\mathbb{P}, \mathbb{P}'$, where $\mathbb{P} \xrightarrow{o} \mathbb{P}'$, the MBPGs of $\mathbb{P}$ and $\mathbb{P}'$ are identical.

Definition

A set of Namos N is called MBPG-complete, if for all pairs of pangenomes $\mathbb{P}, \mathbb{P}'$ with the same MBPG there is a sequence of Namos $o_1 o_2 \ldots o_k \in N^*$, such that $\mathbb{P} \xrightarrow{o_1}\xrightarrow{o_2} \cdots \xrightarrow{o_k} \mathbb{P}'$.

SCJ - A simple Amo



SCJ-CARP Theoretical result

- SCJ Large Parsimony is NP-hard

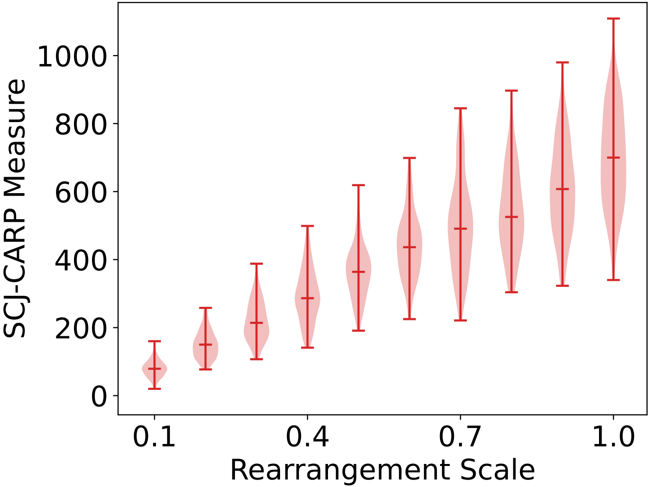
SCJ-CARP Theoretical result

- SCJ Large Parsimony is NP-hard
- However SCJ-CARP:

Lemma

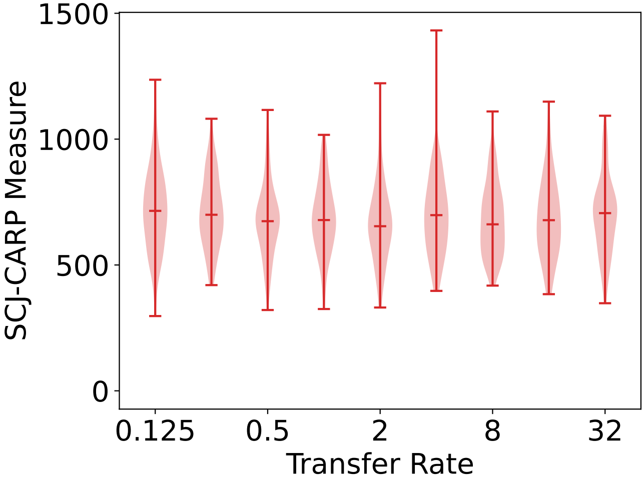
One needs exactly $|E_C|$ SCJs to transform a pangenome into a simple one where $E_C \subset E_A$ is a subset of adjacency edges that can be determined in linear time.

SCJ-CARP Tracks Rearrangements



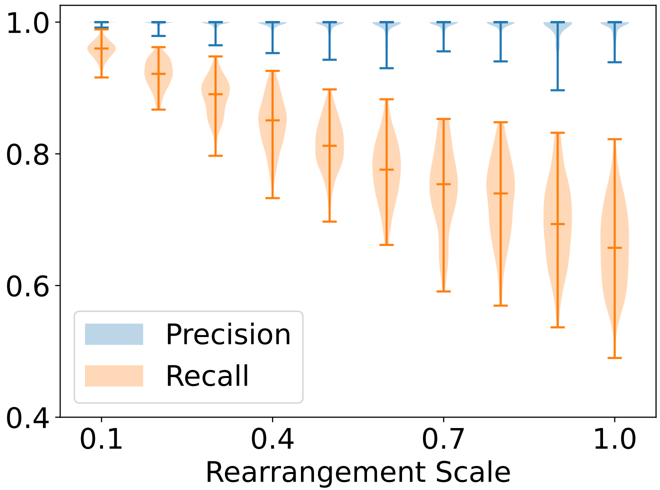
Pearson Coefficient: 0.88

SCJ-CARP Does Not Track Horizontal Effects



Pearson Coefficient: -0.01

SCJ-CARP Reconstructs Ancestral Adjacencies (to a Limited Extent)



CARP- Long Term Perspectives

- Tree/Lineages implied by CARP scenario

CARP- Long Term Perspectives

- Tree/Lineages implied by CARP scenario
- CARP for more complex rearrangement models (DCJ/HP/BI...)

CARP- Long Term Perspectives

- Tree/Lineages implied by CARP scenario
- CARP for more complex rearrangement models (DCJ/HP/BI...)
  - → Characterize Pangenome by vector of CARP measures $(m_{SCJ}, m_{DCJ}, m_{HP}, m_{BI}, \ldots)$.
  - → Compare vectors to compare pangenomes

CARP- Long Term Perspectives

- Tree/Lineages implied by CARP scenario
- CARP for more complex rearrangement models (DCJ/HP/BI...)
  - → Characterize Pangenome by vector of CARP measures $(m_{SCJ}, m_{DCJ}, m_{HP}, m_{BI}, \ldots)$.
  - → Compare vectors to compare pangenomes
- Comparing pangenomes $\mathbb{P}_a, \mathbb{P}_b$ on the same marker set
  - Among all possible CARP ancestors A_a, A_b
  - $d(\mathbb{P}_a, \mathbb{P}_b) = \min_{\mathbb{A}_a \in A_a, \mathbb{A}_b \in A_b} d(\mathbb{A}_a, \mathbb{A}_b)$

CARP- Long Term Perspectives

- Tree/Lineages implied by CARP scenario
- CARP for more complex rearrangement models (DCJ/HP/BI...)
  - → Characterize Pangenome by vector of CARP measures $(m_{SCJ}, m_{DCJ}, m_{HP}, m_{BI}, \ldots)$.
  - → Compare vectors to compare pangenomes
- Comparing pangenomes $\mathbb{P}_a, \mathbb{P}_b$ on the same marker set
  - Among all possible CARP ancestors A_a, A_b
  - $d(\mathbb{P}_a, \mathbb{P}_b) = \min_{\mathbb{A}_a \in A_a, \mathbb{A}_b \in A_b} d(\mathbb{A}_a, \mathbb{A}_b)$
- Weighted Namos

Conclusion

- Rearrangements are interesting both from a computational and from a biological perspective.

Conclusion

- Rearrangements are interesting both from a computational and from a biological perspective.

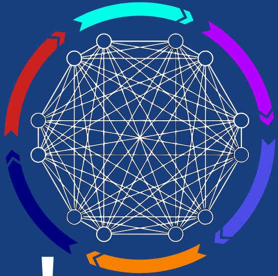
- Simple changes to a problem can make it (computationally) a lot easier.

Conclusion

- Rearrangements are interesting both from a computational and from a biological perspective.
- Simple changes to a problem can make it (computationally) a lot easier.
- There is a wealth of theoretical results for rearrangements that only need slight adaptations to make them useful in practice.

Conclusion

- Rearrangements are interesting both from a computational and from a biological perspective.
- Simple changes to a problem can make it (computationally) a lot easier.
- There is a wealth of theoretical results for rearrangements that only need slight adaptations to make them useful in practice.
- Future work: (Formal definition for) Marker segmentation on pangenomes



Thank you!

slides