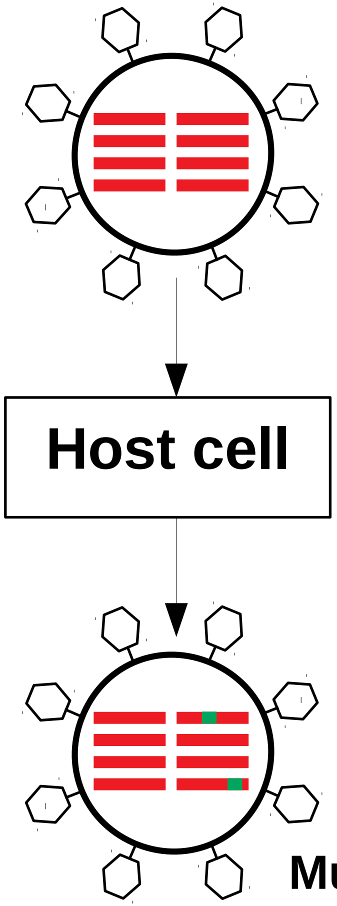
TreeKnit
Inferring Ancestral Reassortment Graphs of influenza viruses

Pierre Barrat-Charlaix

Team of Richard Neher

Evolution of influenza: Mutations and reassortment

Mutation



Flu virus

🟥 Gene : 8 in total

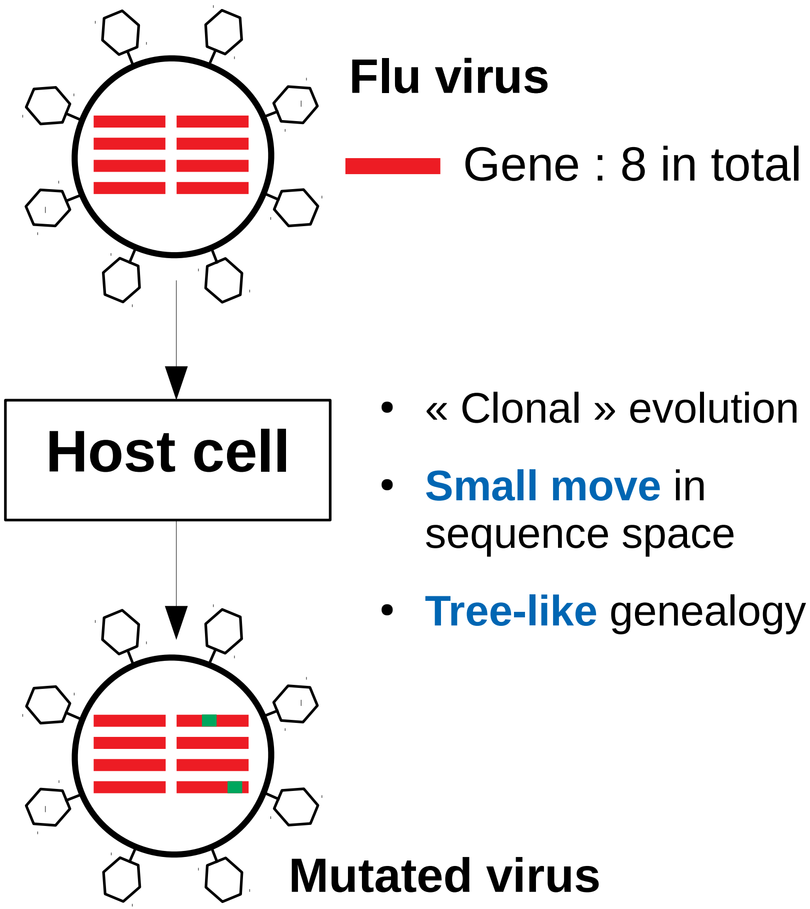
Host cell

Mutated virus

- « Clonal » evolution
- **Small move** in sequence space
- **Tree-like** genealogy

Evolution of influenza: Mutations and reassortment

Mutation



Flu virus

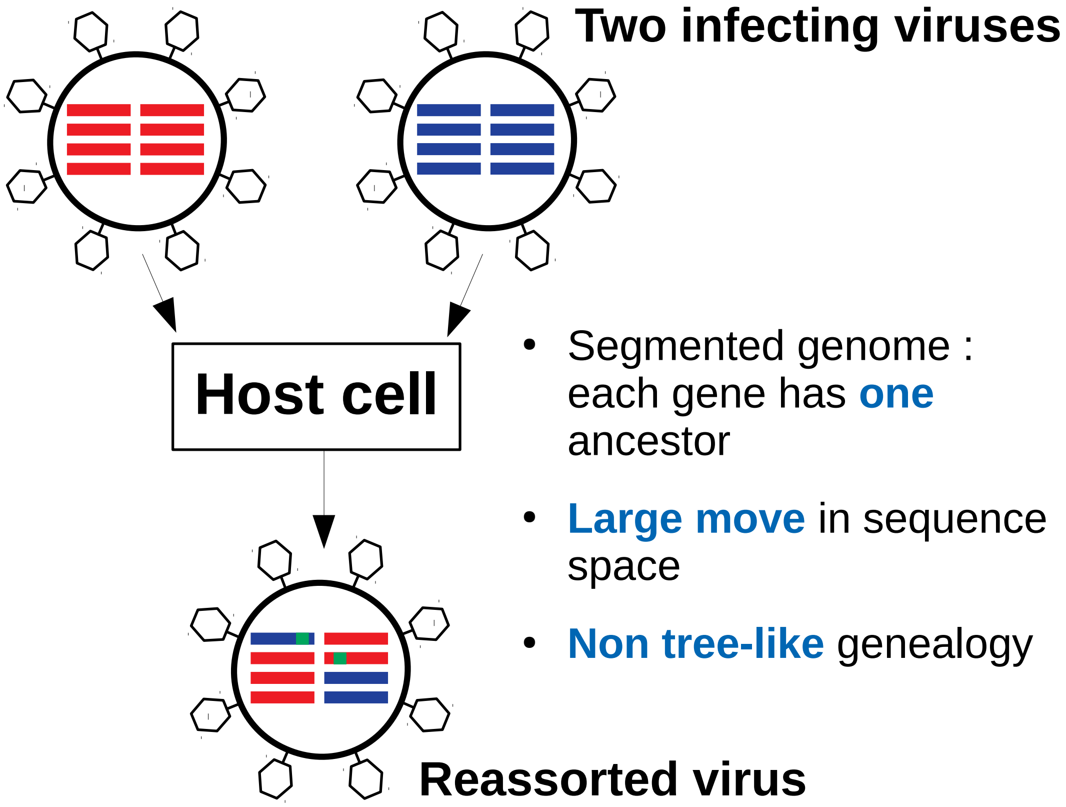
🟥 Gene : 8 in total

Host cell

Mutated virus

- « Clonal » evolution
- **Small move** in sequence space
- **Tree-like** genealogy

Reassortment



Two infecting viruses

Host cell

Reassorted virus

- Segmented genome : each gene has **one** ancestor
- **Large move** in sequence space
- **Non tree-like** genealogy

Reassortment in influenza

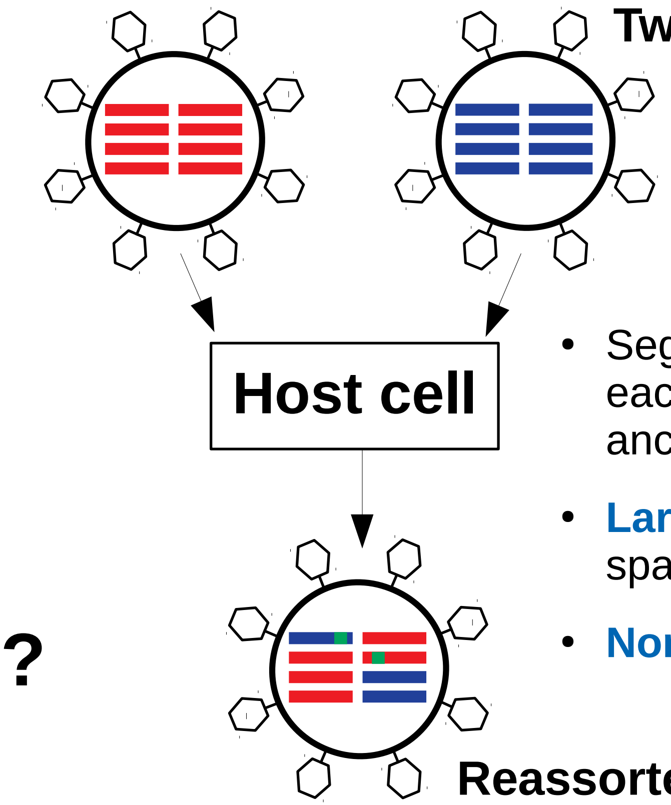
- Combines strains from **different subtypes**, or from **human/animal** hosts.
- Origin of many **pandemics**
  - Asian flu – 1957
  - Hong Kong flu – 1968
  - H1N1 pandemic – 2009
- Also happens at “smaller” scale: within a subtype.

- How often does it happen?
- Contribution to immune escape and adaptation?

→ **?**

Reassortment

Two infecting viruses

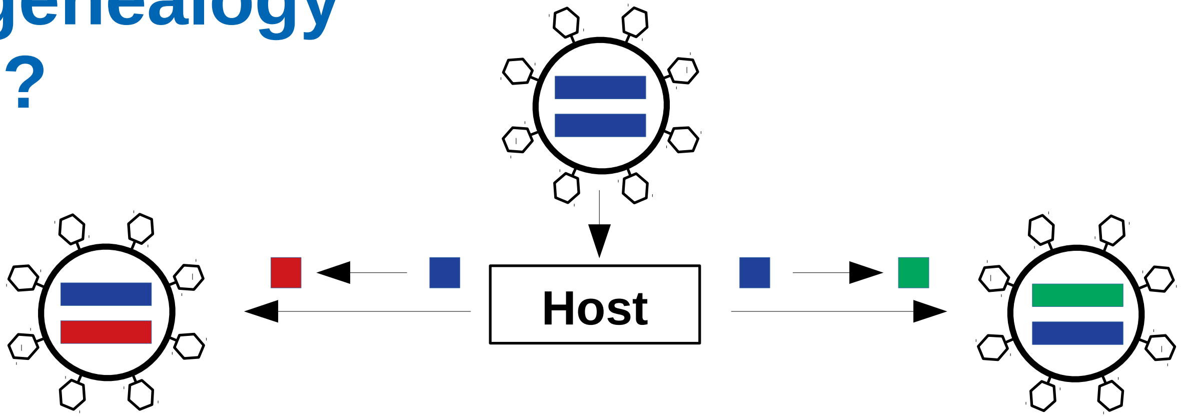


Reassorted virus

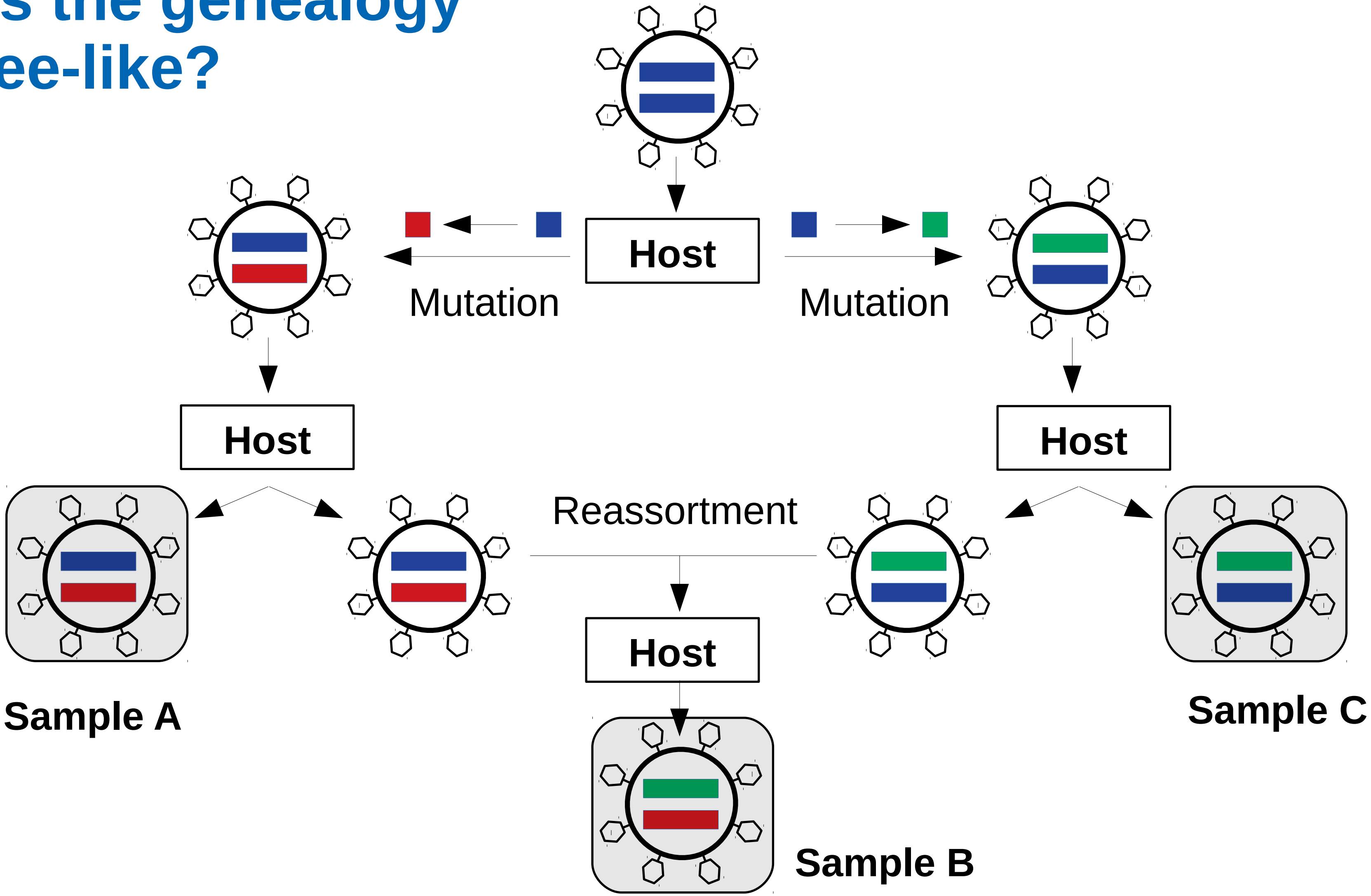
- Segmented genome : each gene has **one** ancestor
- **Large move** in sequence space
- **Non tree-like** genealogy

Reassortments are hard to infer from sequences!

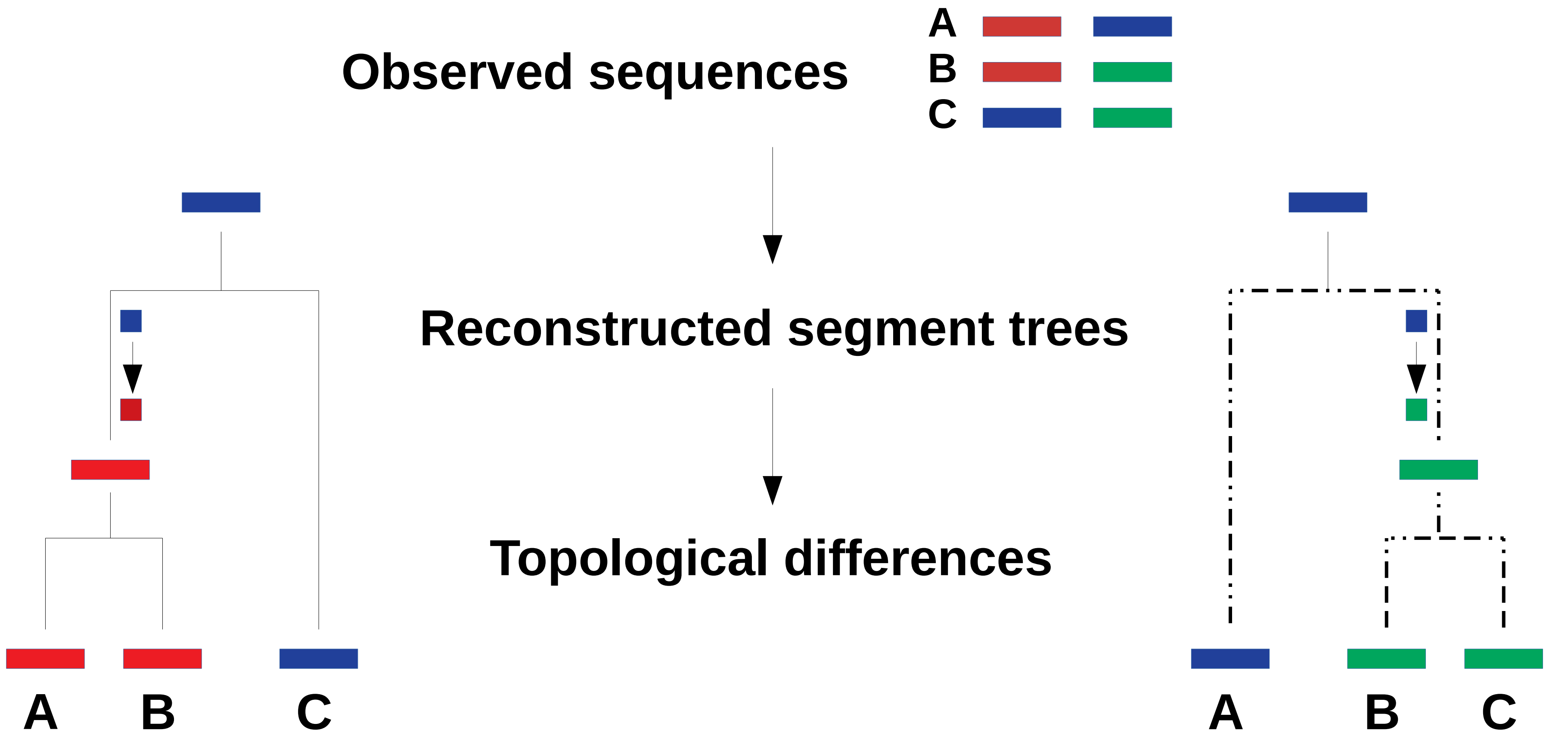
Why is the genealogy not tree-like?



Why is the genealogy not tree-like?



Ancestral Reassortment Graph



Observed sequences

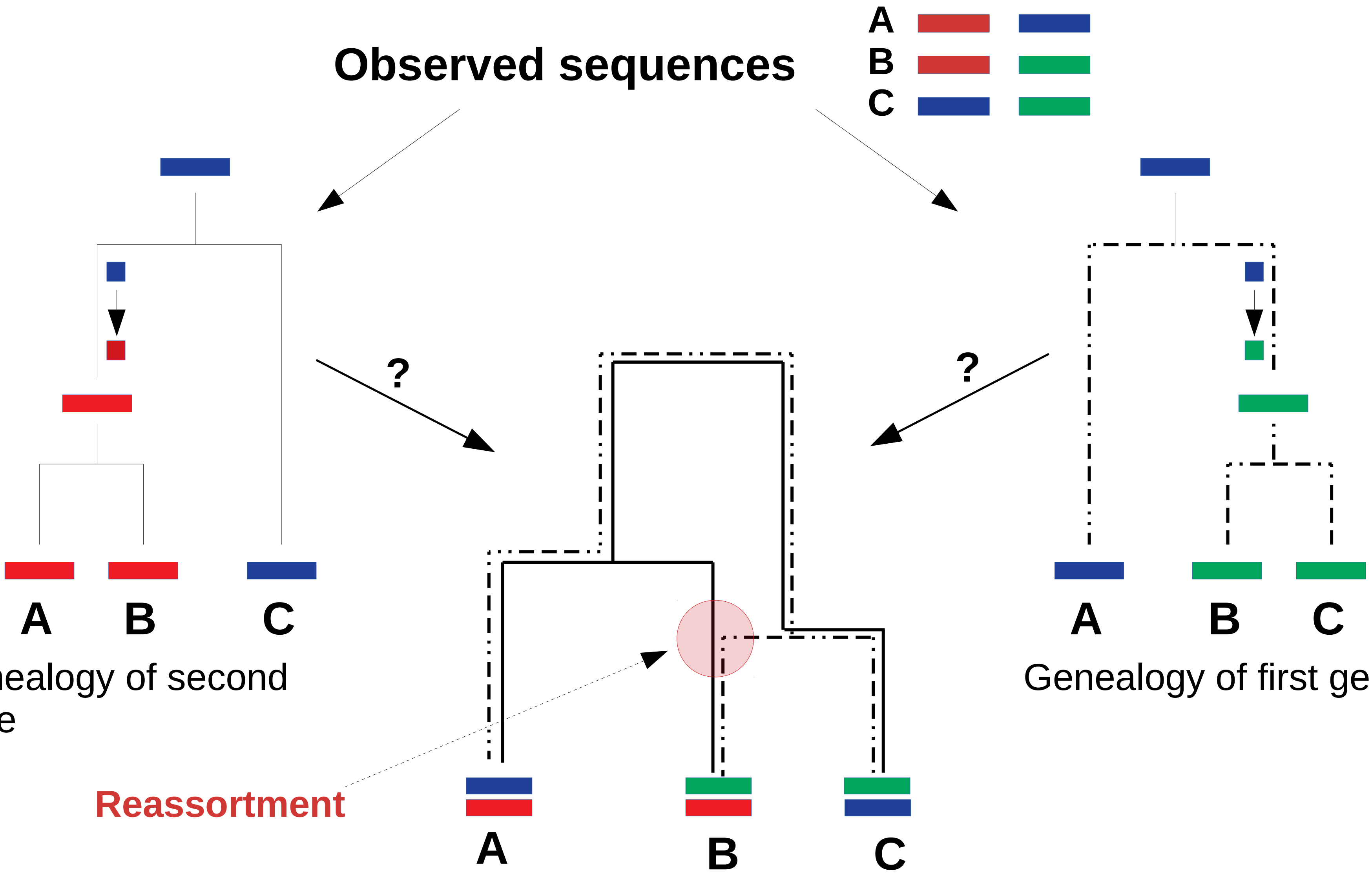
Reconstructed segment trees

Topological differences

Genealogy of first gene

Genealogy of second gene

Why is the genealogy not tree-like?



Observed sequences

Genealogy of second gene

Genealogy of first gene

Reassortment

Ancestral Recombination Graph (ARG)

Inferring reassortments / Reconstructing the ARG

Existing methods

- Manual inspection of trees
  (*e.g.* **[Holmes et. al. 2005], [Boni et. al. 2010]**)
- Methods based on genetic distance **[Rabadan et. al. 2008]**
- Trees + mutation methods [**Villa & Lässig 2017]**
- Tree topology based methods **[Nagarajan & Kingsford 2011]**

→ Finds a subset of reassortment events

- Bayesian methods **[Müller et. al. 2020]** → Accurate but slow

→ **No "reference" method**

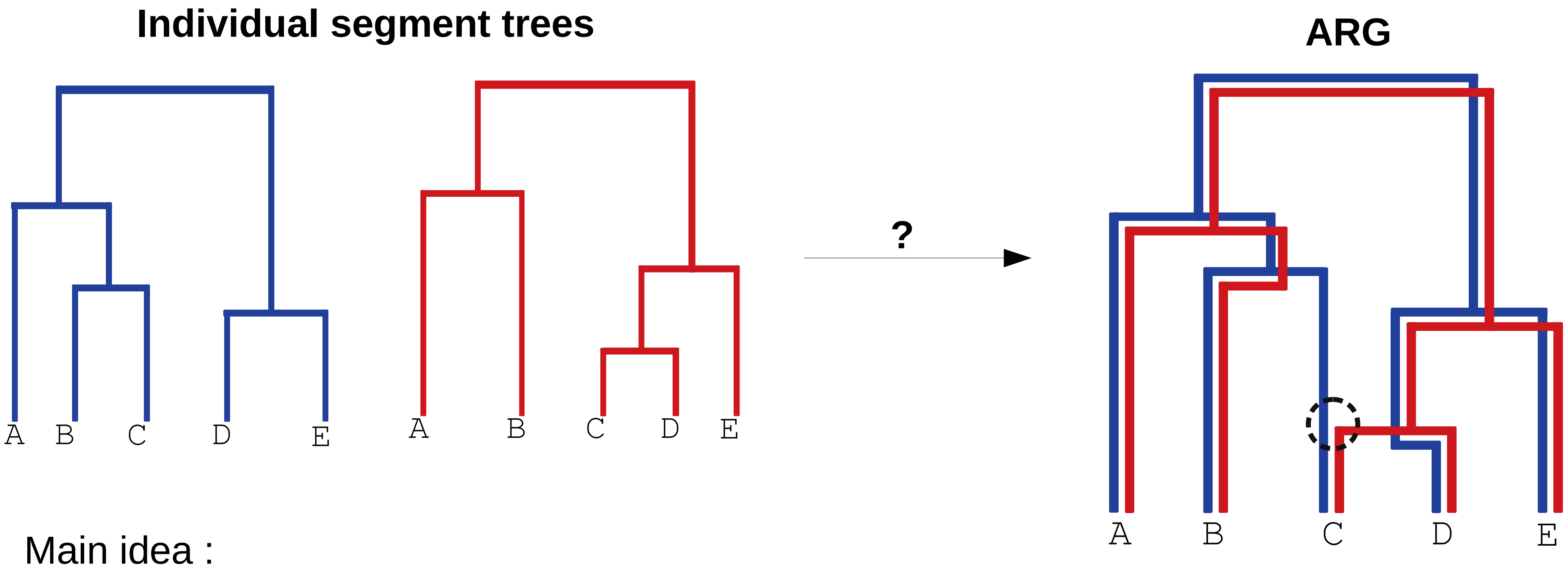
We want something that is

- **Fast** : can be easily applied to new sequences
- Finds **all reassortments**, and not only large obvious ones
- Works for the 2-genes case (simplicity)

Inferring the ARG: the Treeknit method

🟦 Tree of segment 1

🟥 Tree of segment 2



Individual segment trees

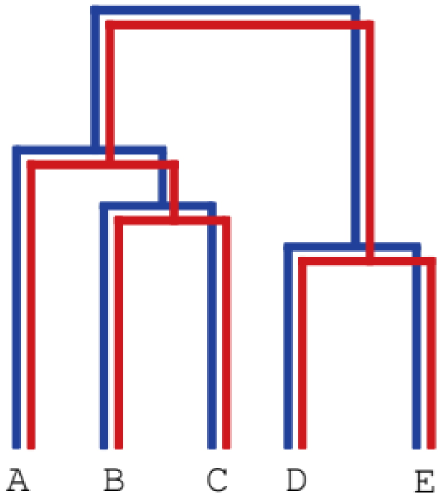
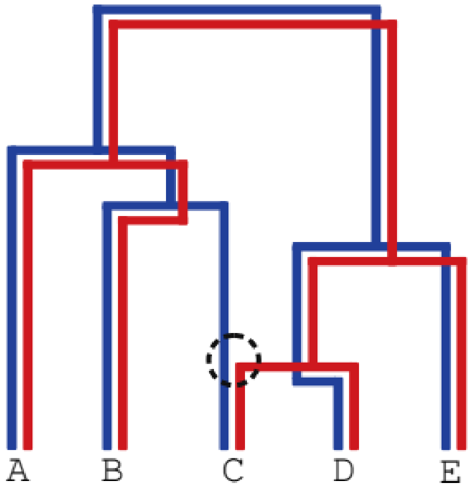
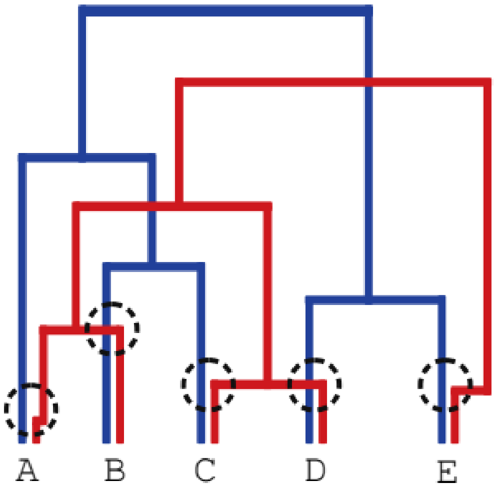
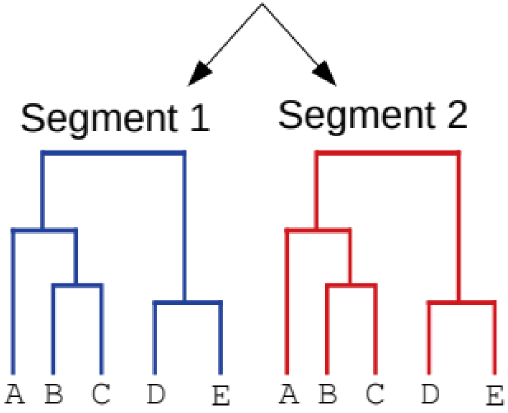
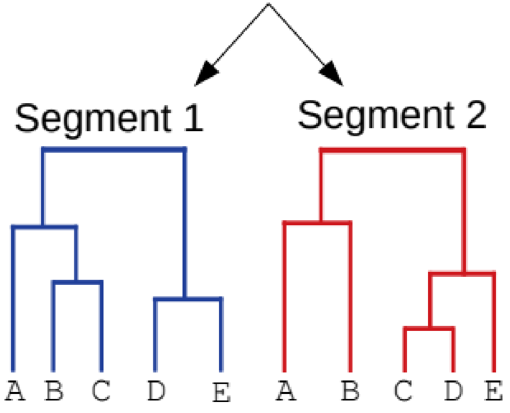
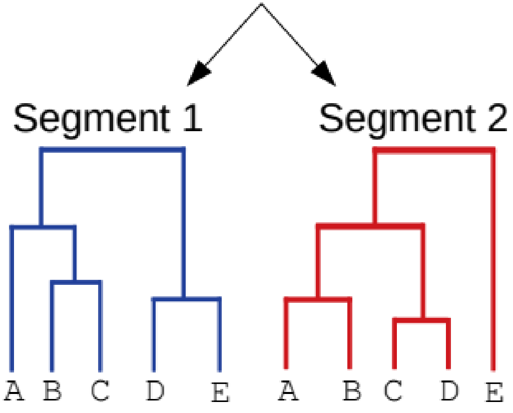
ARG

Main idea :

- The ARG is a **collage of gene trees**
- We can **infer each tree** from sequences (iqtree, RaxML, …)
- **Topological differences** between these trees are **due to reassortment**

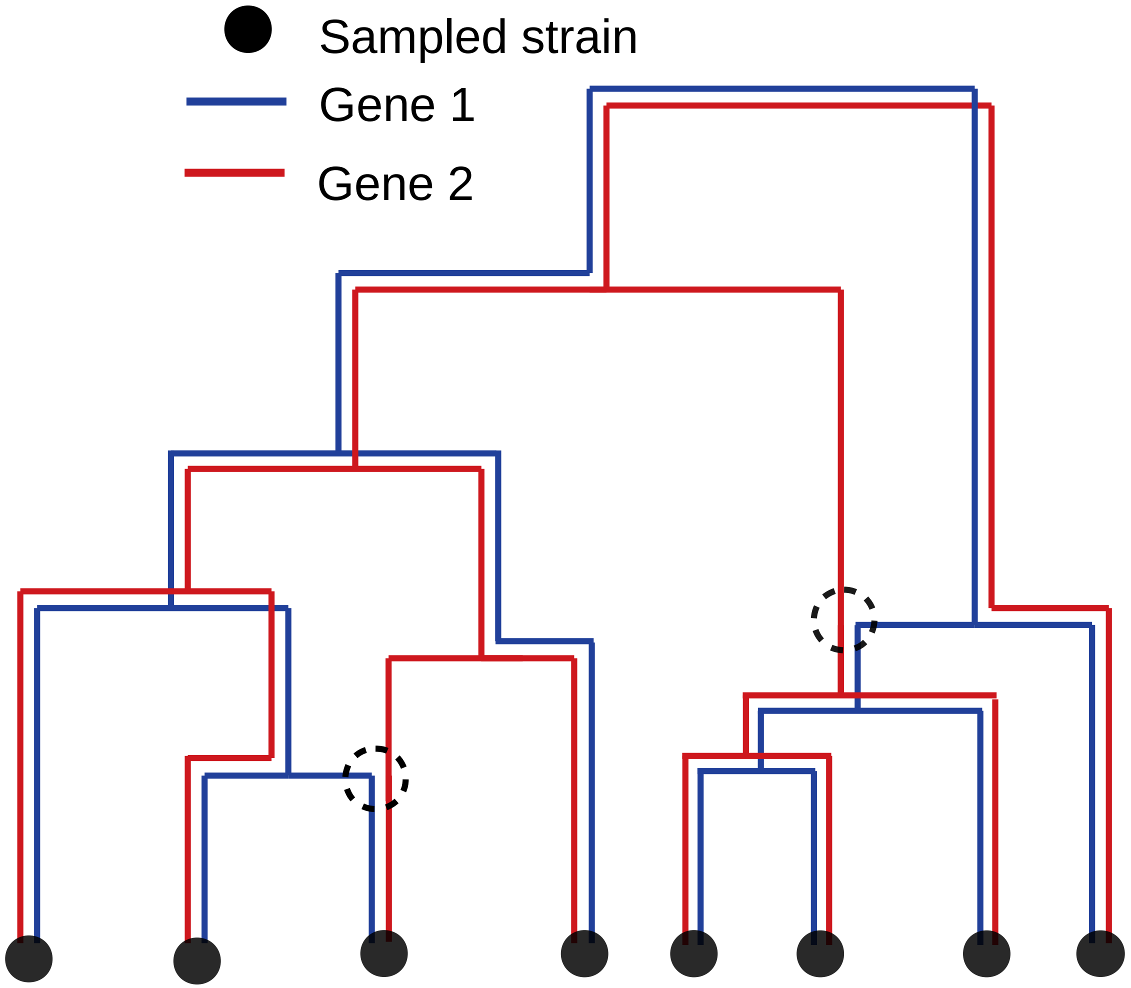
→ **Method based on topological differences between trees**

Inferring the ARG

	Low $r^\star$	Intermediate $r^\star$	High $r^\star$
ARG			
Observed segment trees			
Difficulty	Easy	Hard	Easy

Maximally compatible clades (MCCs)

The ARG is a **collage of gene trees**



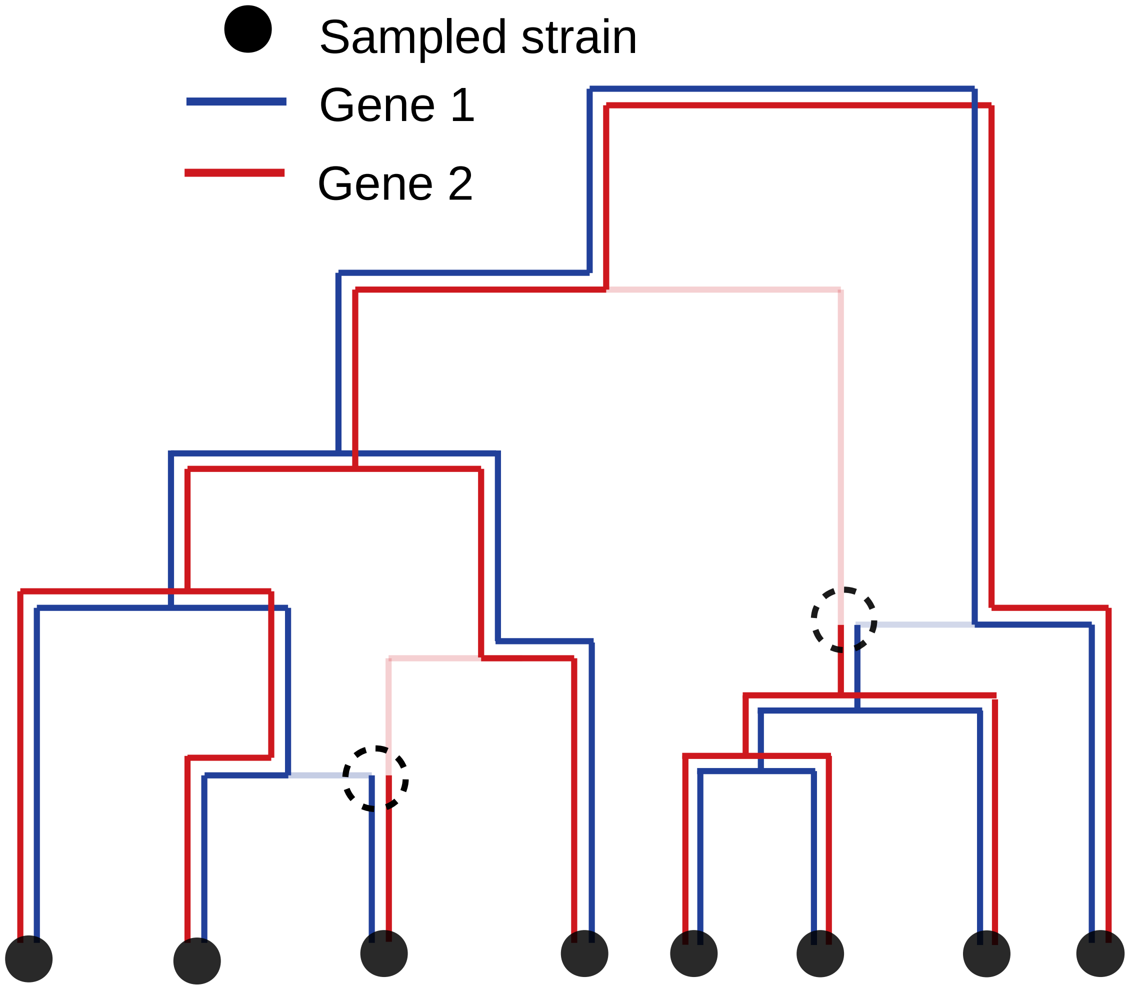
Maximally compatible clades (MCCs)

The ARG is a **collage of gene trees**

Restricting to branches that
belong to both trees

↓

Maximally compatible clades

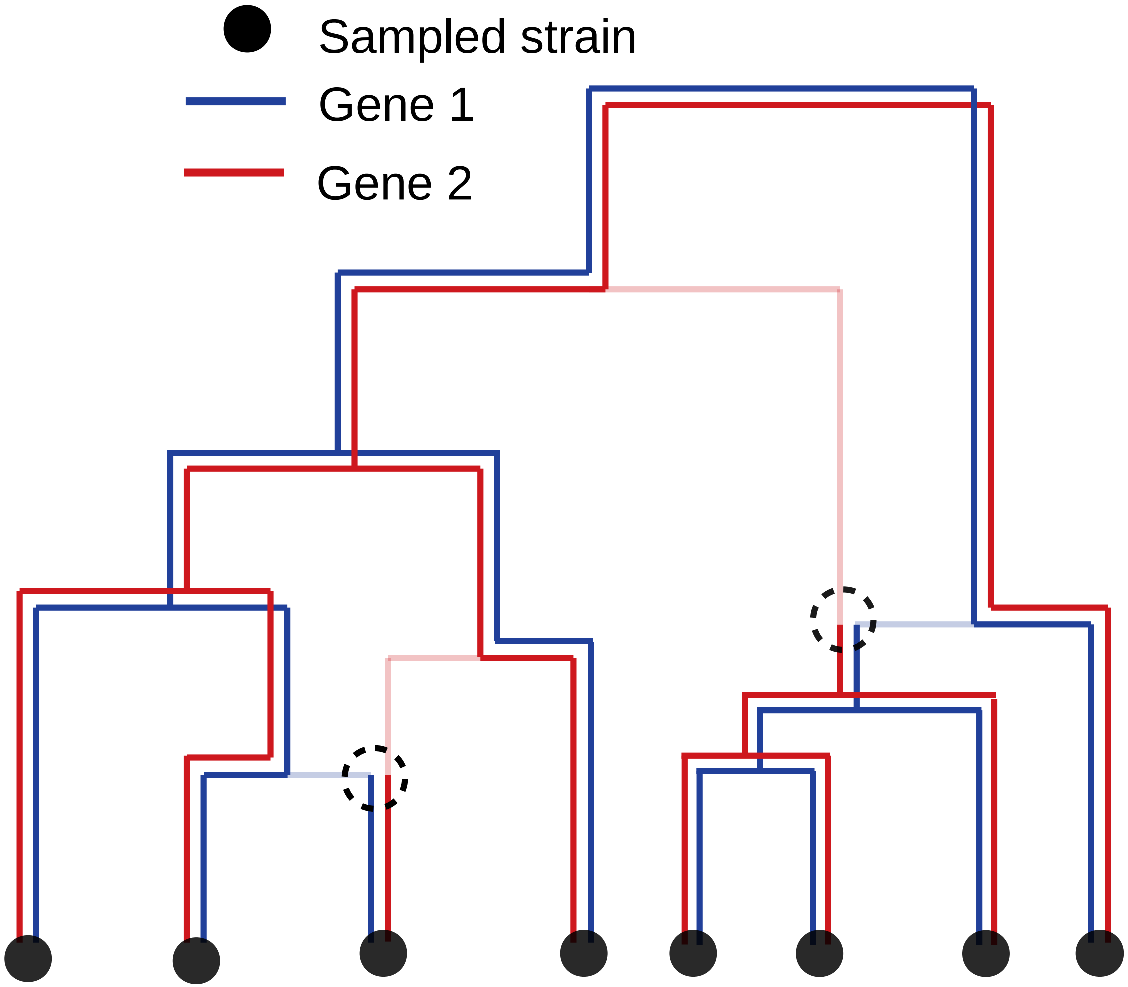


Maximally compatible clades (MCCs)

The ARG is a **collage of gene trees**

Restricting to branches that **belong to both trees**

Maximally compatible clades

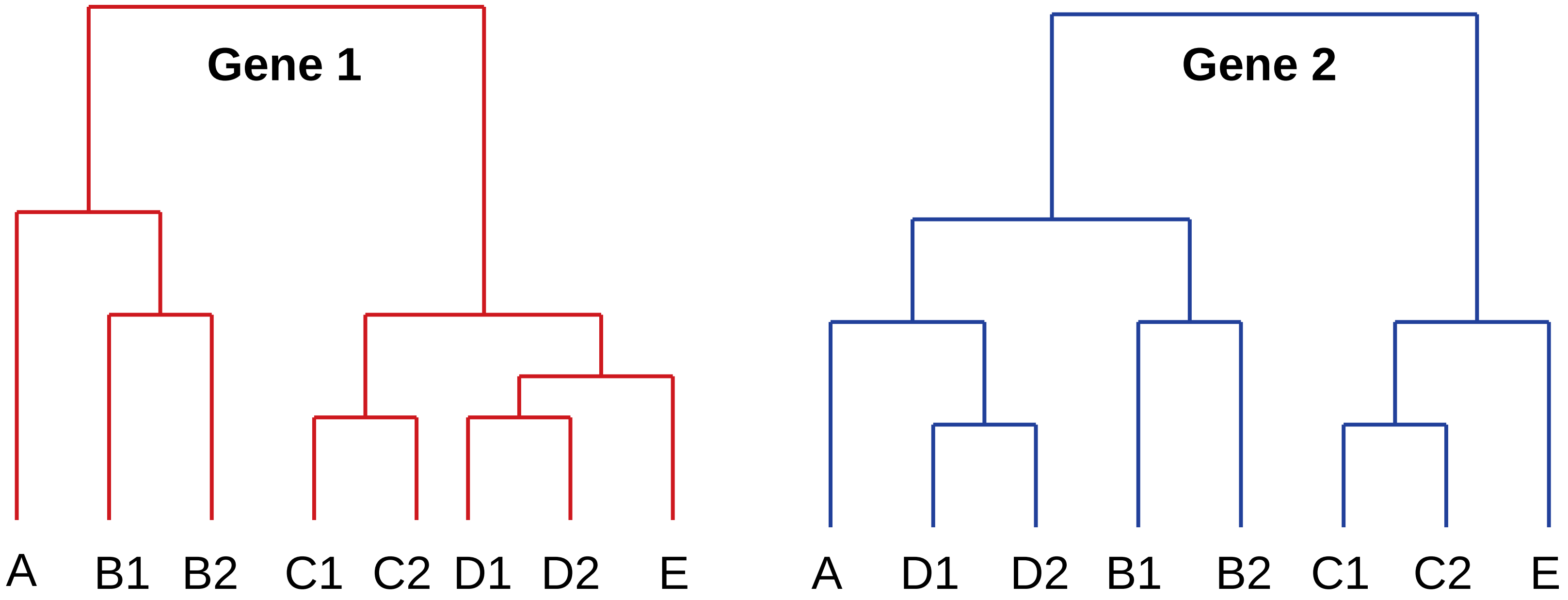


- The **root of an MCC** is either
  - A reassortment
  - The root of both trees
- If **both trees** and **all MCCs** are known, then the **ARG** is known

Inferring the ARG → Inferring MCCs

First step: naive estimation of MCCs

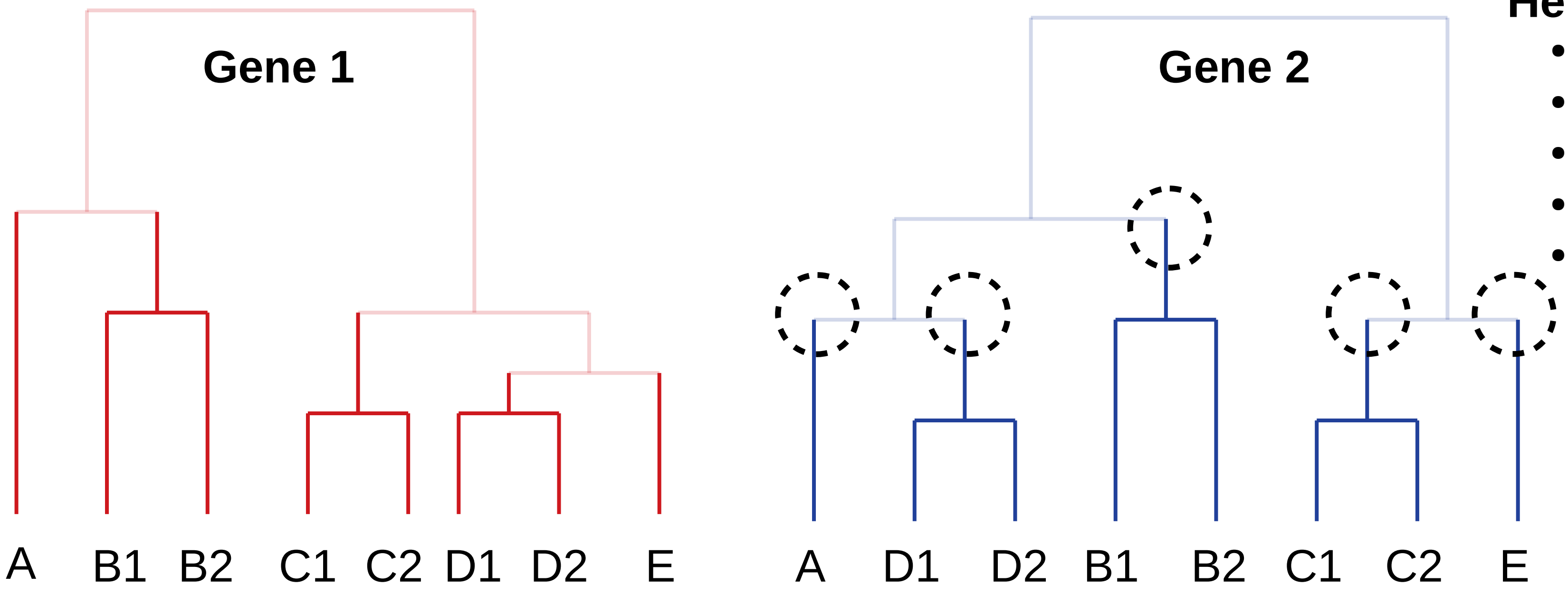
→ Take clades that have exactly matching topologies



Inferring the ARG ⟶ Inferring MCCs

First step: naive estimation of MCCs

⟶ Take clades that have exactly matching topologies



Here : 5 naive MCCs

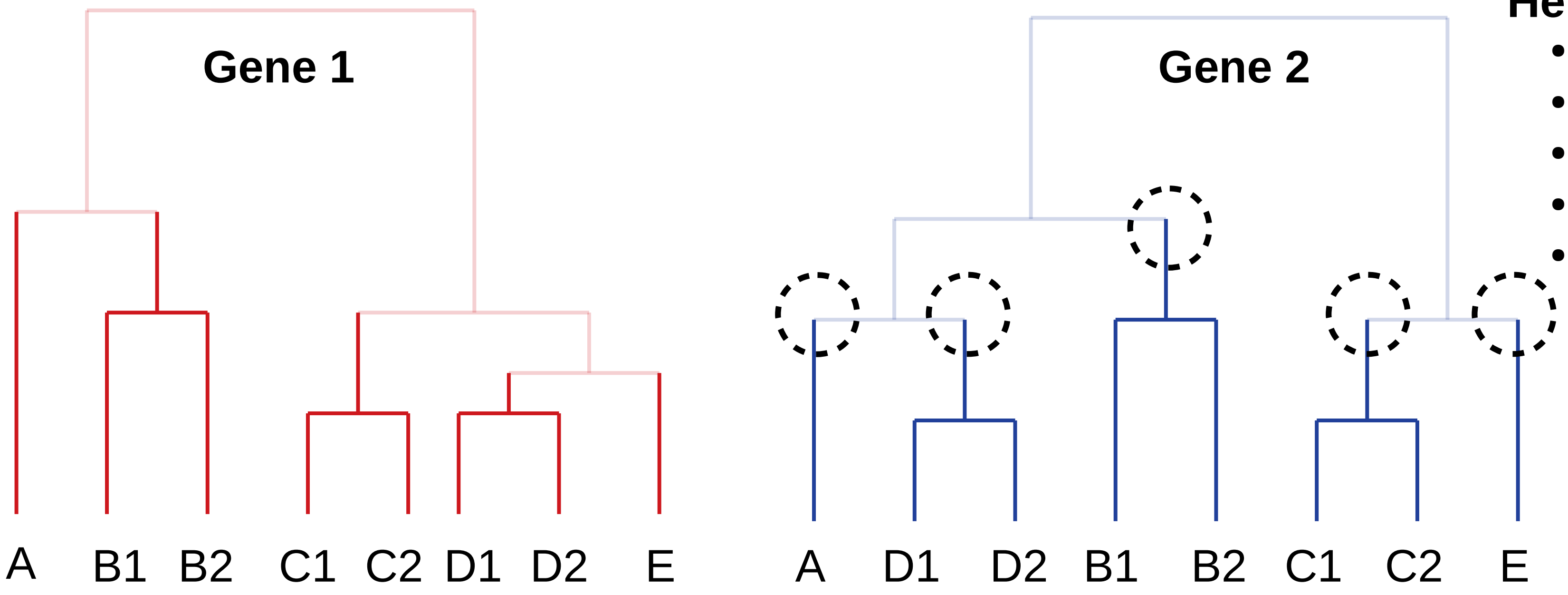
- A
- B1, B2
- C1, C2
- D1, D2
- E

5 reassortments !

Inferring the ARG → Inferring MCCs

First step: naive estimation of MCCs

→ Take clades that have exactly matching topologies



Here : 5 naive MCCs

- A
- B1, B2
- C1, C2
- D1, D2
- E

↓

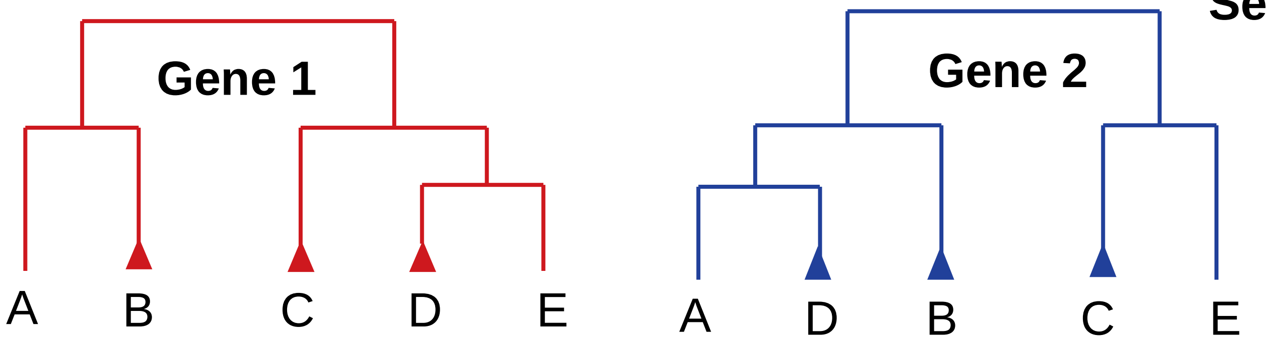
5 reassortments !

Naive estimation :

Finds too many MCCs → Too many reassortments

Conservative approach → Does not overextend MCCs

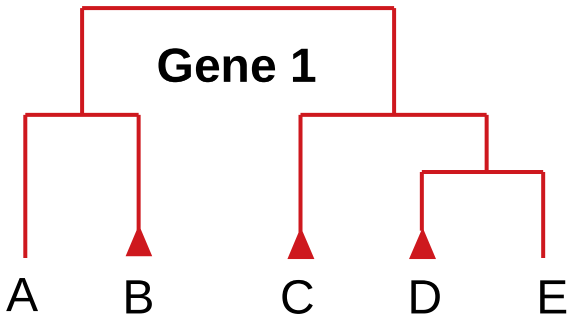
Inferring MCCs

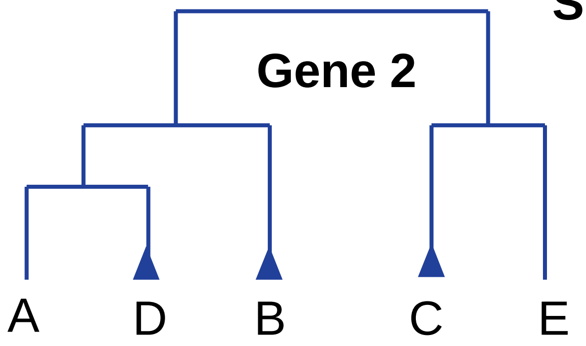


Second step: “reduce” to naive MCCs

- (B1, B2) → B
- (C1, C2) → C
- (D1, D2) → D

Inferring MCCs: Parsimonious approach





Second step: “reduce” to naive MCCs

- (B1, B2) → B
- (C1, C2) → C
- (D1, D2) → D

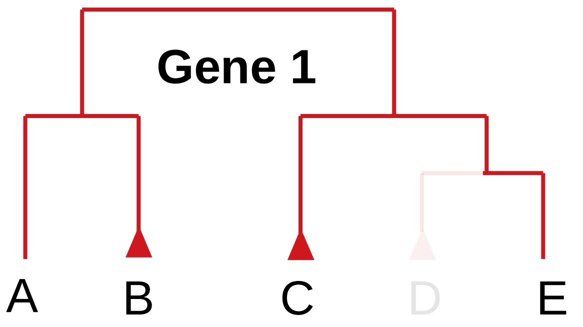
By eye:
D is the reassorted clade.
How can we **formalize** this?

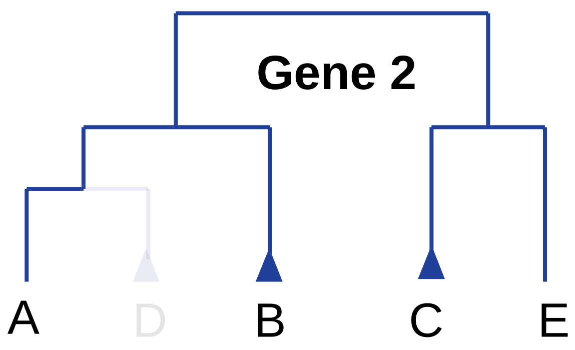
Surrounding of each leaf: **clade** defined by parent:

- A → (A,B) / (A,D)
- B → (A,B) / (A,D,B)
- C → (C,D,E) / (C,E)
- D → (D,E) / (A,D)
- E → (D,E) / (C,E)

→ **5 incompatibilities**

Inferring MCCs: Parsimonious approach





First step: “reduce” to naive MCCs

- (B1, B2) → B
- (C1, C2) → C
- (D1, D2) → D

By eye:
D is the reassorted clade.
How can we **formalize** this?

Surrounding of each leaf: **clade** defined by parent:

- A → (A,B) / (A,D)
- B → (A,B) / (A,D,B)
- C → (C,D,E) / (C,E)
- D → (D,E) / (A,D)
- E → (D,E) / (C,E)

→ **5 incompatibilities**

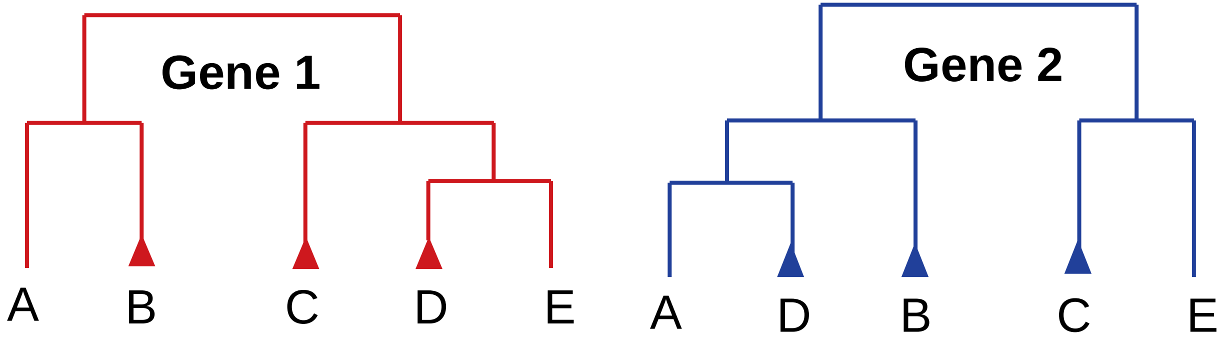
Hypothesis: D is a reassortant → Remove it from the trees

- A → (A,B) / (A,B)
- B → (A,B) / (A,B)
- C → (C,E) / (C,E)
- ~~D → (D,E) / (A,D)~~
- E → (C,E) / (C,E)

→ **0 incompatibilities**

0 remaining reassortments!

Inferring MCCs



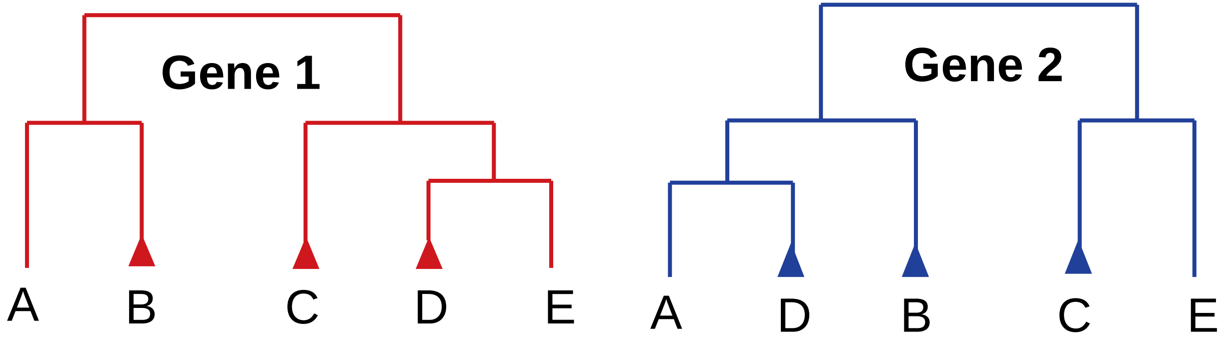
For each leaf ***n***

- σ_n
  - 1 if we **remove** ***n***
  - 0 otherwise

- $\vec{\sigma} = (\sigma_1 \ldots \sigma_L)$: “configuration” vector

- $\Delta(n, \vec{\sigma})$
  - 1 if **incompatibility** above ***n***
  - 0 otherwise

Inferring MCCs



For each leaf ***n***

→ σ_n

1 if we **remove *n***

0 otherwise

→ $\vec{\sigma} = (\sigma_1 \ldots \sigma_L)$: “configuration” vector

→ $\Delta(n, \vec{\sigma})$

1 if **incompatibility** above ***n***

0 otherwise

# of removed leaves

# of incompatibilties

Minimize $N_\gamma(\vec{\sigma}) = \boxed{\sum_{n \in leaves} \Delta(n, \vec{\sigma})\sigma_n} + \gamma\boxed{(L - |\vec{\sigma}|)}$

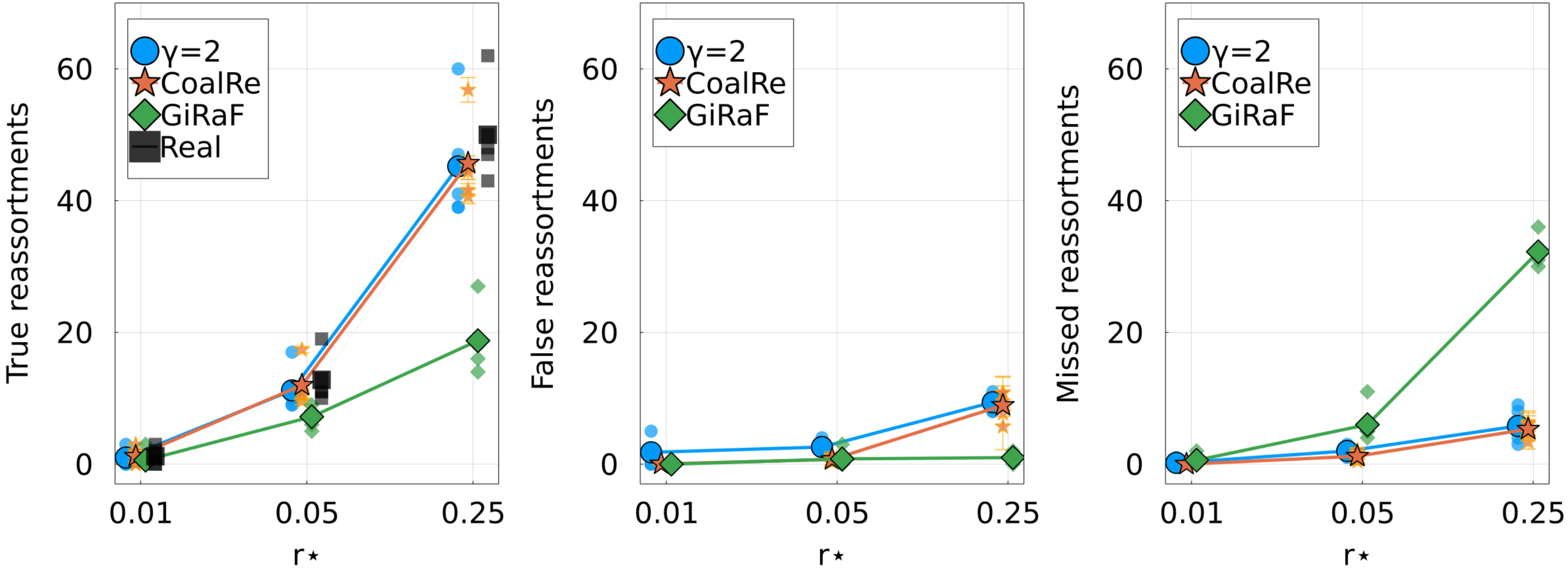
(Simulated annealing)

Minimize incompatibilities with a minimal number of reassortments

Comparison w. other methods

CoalRe: ML based **[Müller et. al. 2020]**

GiRaF: topology based **[Nagarajan & Kingsford 2011]**



Runtime

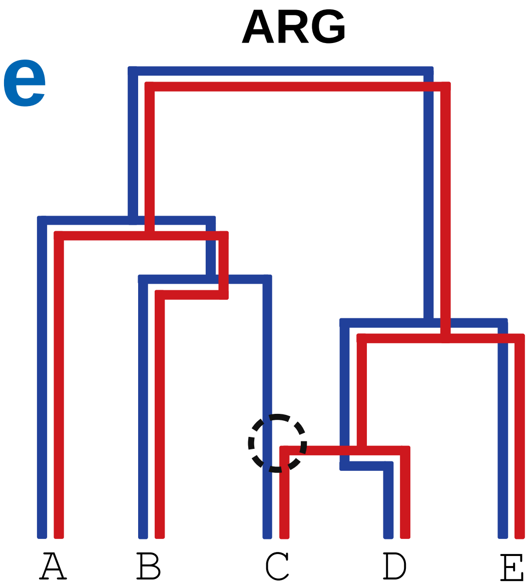
	CoalRe	GiRaF	Treeknit
Inferring trees		20min	30s
Inferring the ARG	~hours	40s	40ms

for 100 leaves

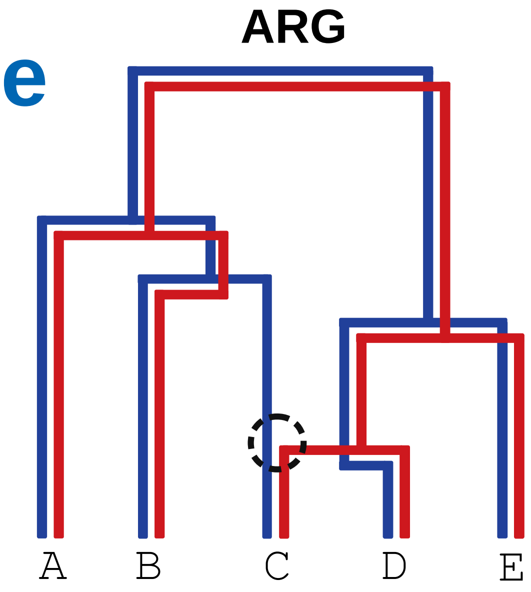
Application: resolving trees, inference

Shared regions of the ARG ⟶ **~ doubled sequence length**

- Better resolved trees
- Better inference of branch length, dates of internal nodes, etc...

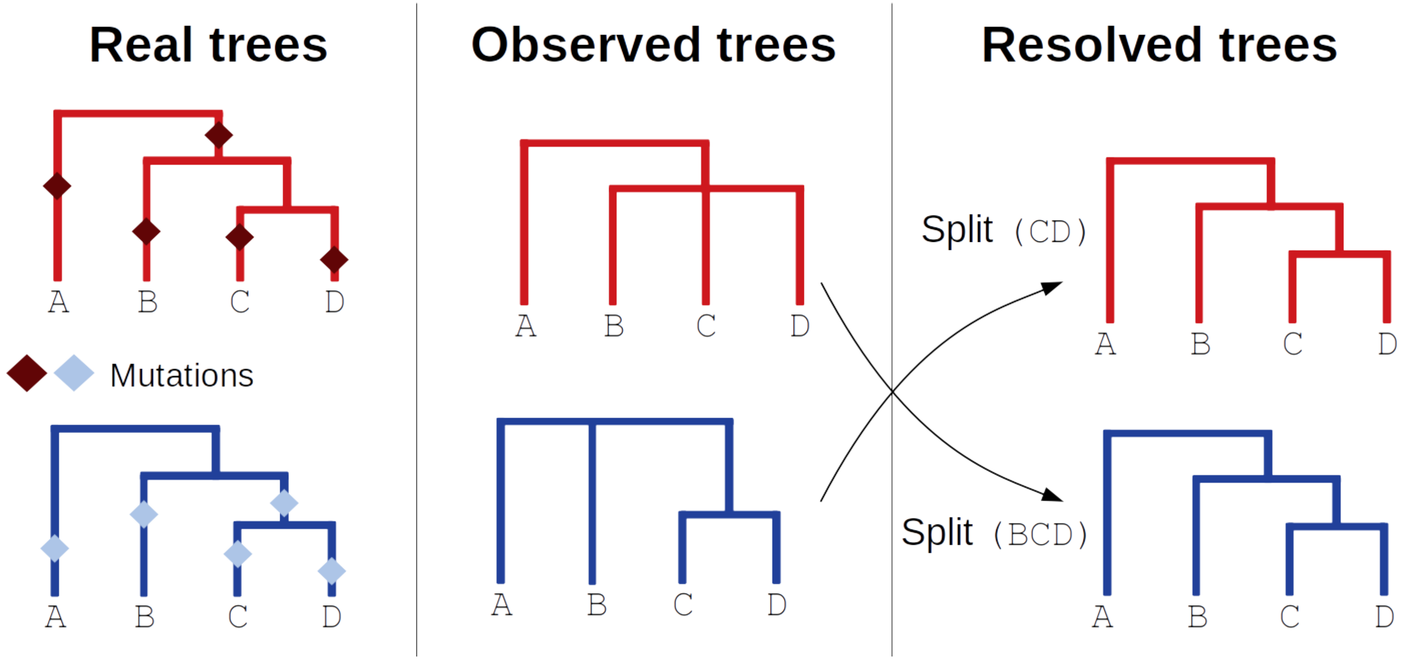


Application: resolving trees, inference

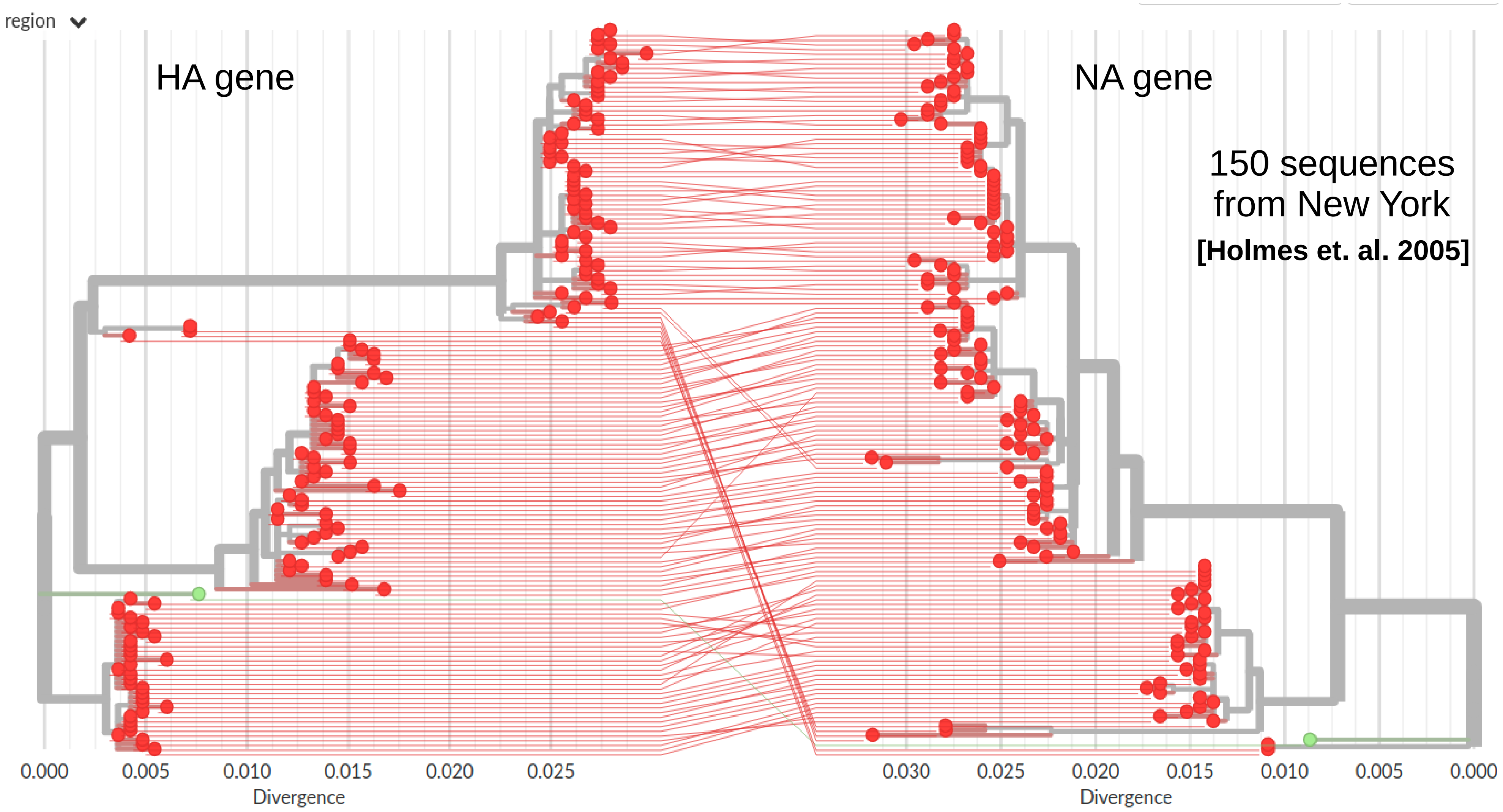


Shared regions of the ARG ⟶ **~ doubled sequence length**

- Better resolved trees
- Better inference of branch length, dates of internal nodes, etc...



Application: disentangling tanglegrams

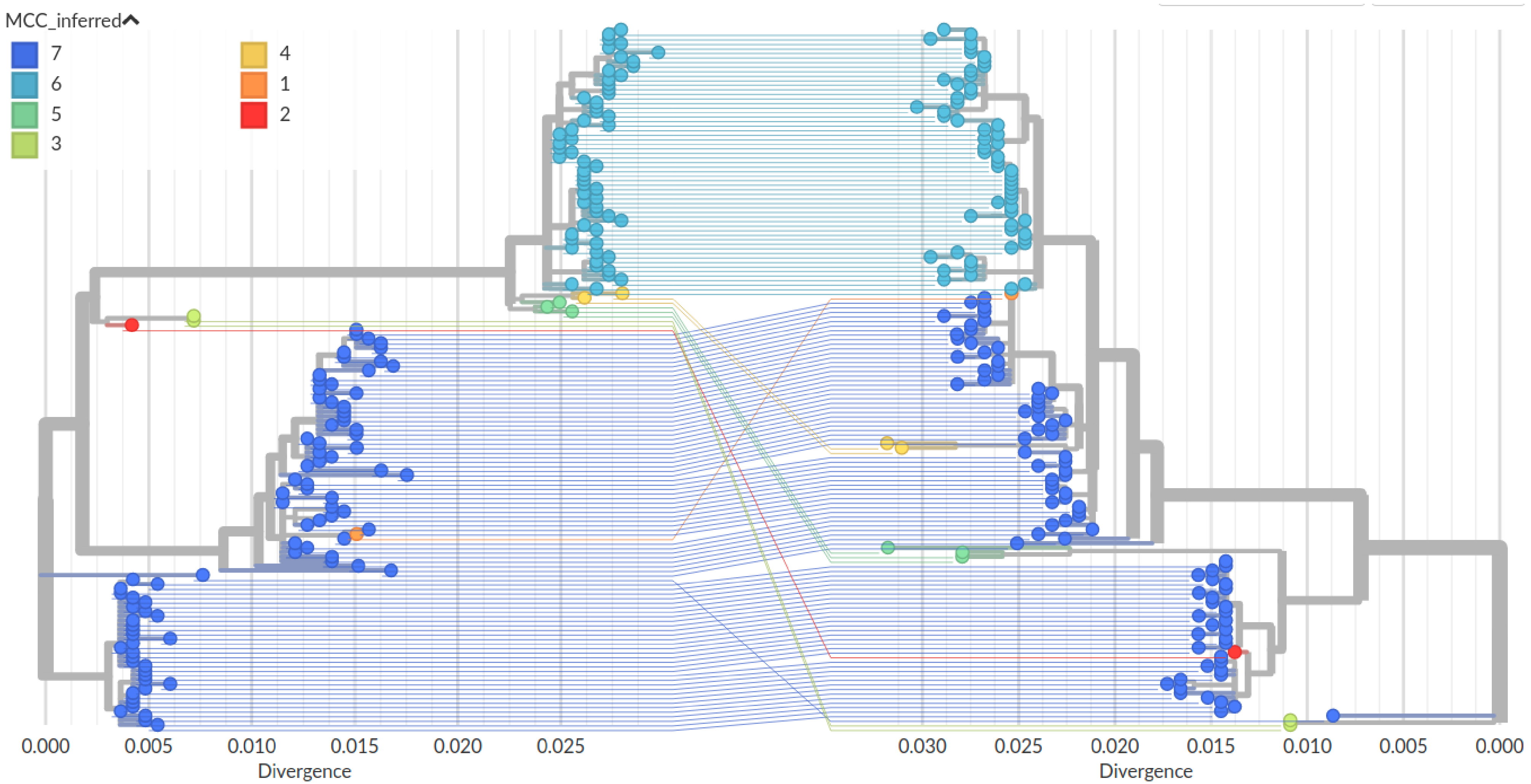


150 sequences from New York

[Holmes et. al. 2005]

Without the knowledge of reassortments: hard problem

Application: disentangling tanglegrams



With the knowledge of reassortments: easy

Summary

Available at **github.com/PierreBarrat/TreeKnit**

Results

- **Treeknit:** Heuristic to infer ARGs from two trees
- **Fast** runtime
- **Good performance** on **simulated data** for all reassortment rates

Applications / challenges

- Resolve trees
- Inference on the ARG
- Visualisation: disentangle tanglegrams
- Knowledge of the ARG → Effect of reassortment on influenza evolution
- Apply to more than two segment trees

Thank you for listening!

Interpretation of gamma

$$N_\gamma(\vec{\sigma}) = \sum_{n \in leaves} \Delta(n, \vec{\sigma})\sigma_n + \gamma(L - |\vec{\sigma}|)$$

- $\gamma \to \infty$ **Infinite cost** for removing leaves → **Naive approach**

- $\gamma = 1$ $N(\vec{\sigma})$ = **# incompatibilities** + **# removed leaves**

  ↑ Reassortments w. naive approach ↑ Enforced reassortments

  $N(\vec{\sigma})$ = **Total number of reassortments** → **"Parsimonious" approach**

- Intermediate γ → **Interpolate** between **naive** and **"parsimonious"**