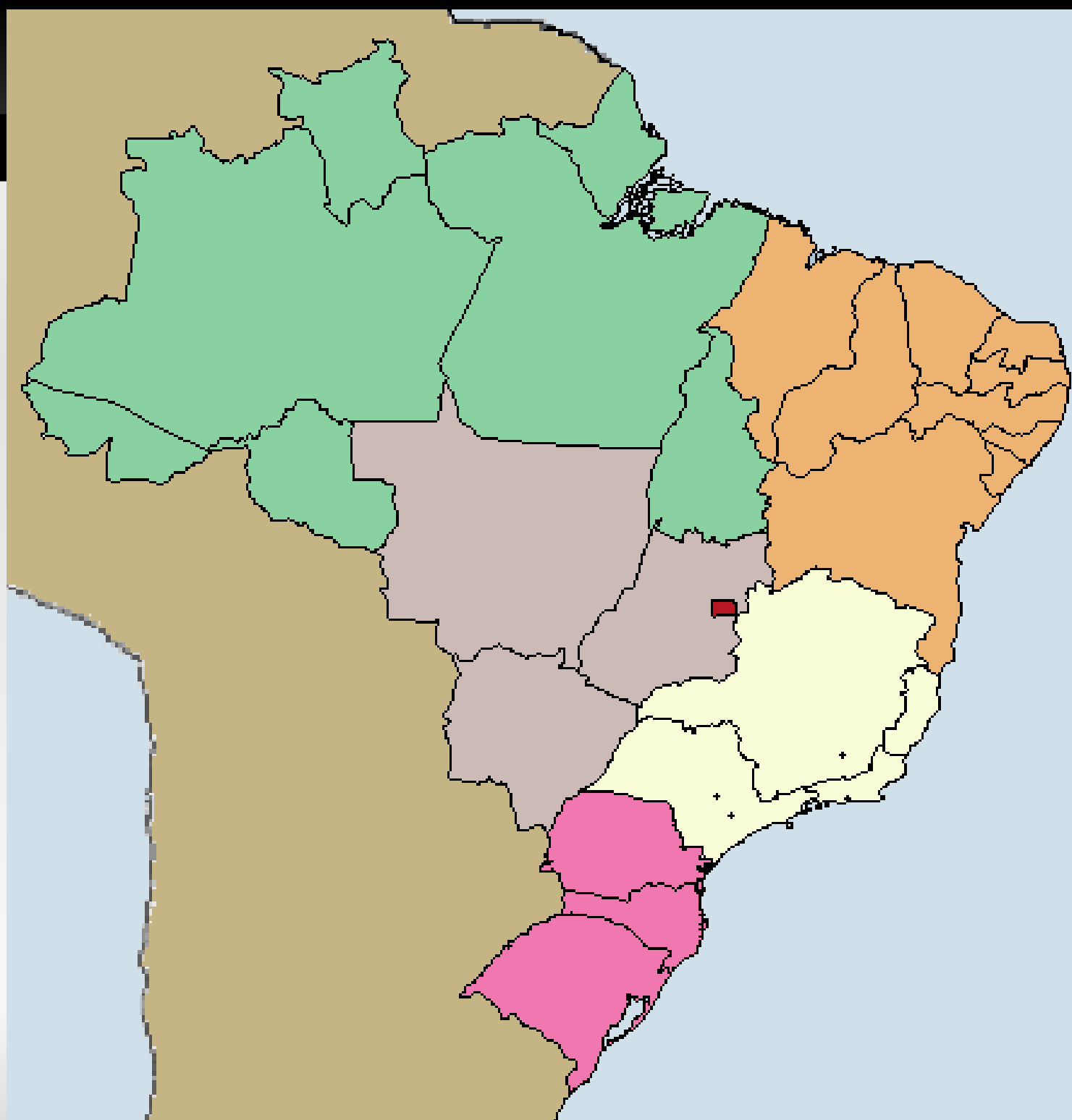


**Exploring the solution  
space of sorting by  
reversals when analyzing  
genome rearrangements**

Marília D. V. Braga

Université Lyon 1 / Programme Alisan / INRIA / CNRS

# Brazil



# Brazil

Juiz de Fora (MG)



Juiz de Fora

Rio de Janeiro

# Brazil



Campinas (SP)  
from 1994 to 2005

**University of Campinas**

**Institute of Computing**

João Meidanis

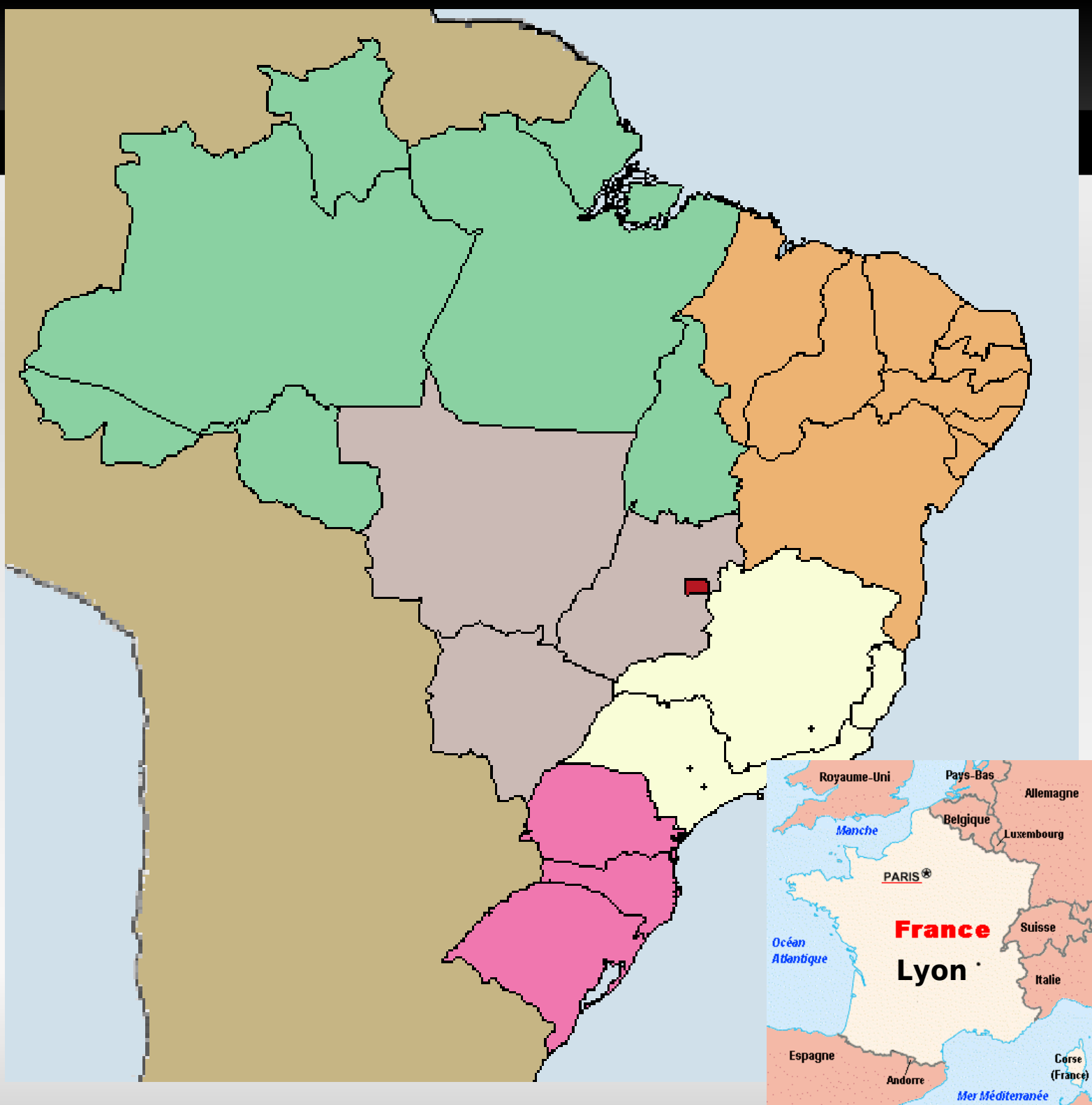
# Brazil, France

Lyon (France)  
from 2005 to 2008

**Université Lyon 1**

**Laboratoire de  
Biométrie et Biologie  
Evolutive**

Marie-France Sagot



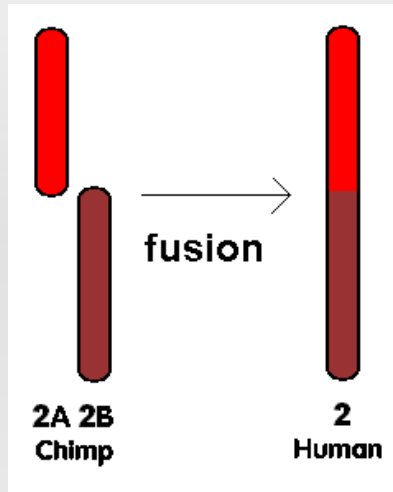
- . **Introduction: genome rearrangements**
- . **Sorting by reversals (SR)**
- . **The solution space of SR**
  - Traces (classes of solutions)
  - Enumerating all the traces
- . **Taking biological constraints in consideration**
  - Common intervals
  - Application: evolution of *Rickettsia* bacterium
- . **baobabLuna**
- . **Conclusions**

**Introduction**

**Genome rearrangements**

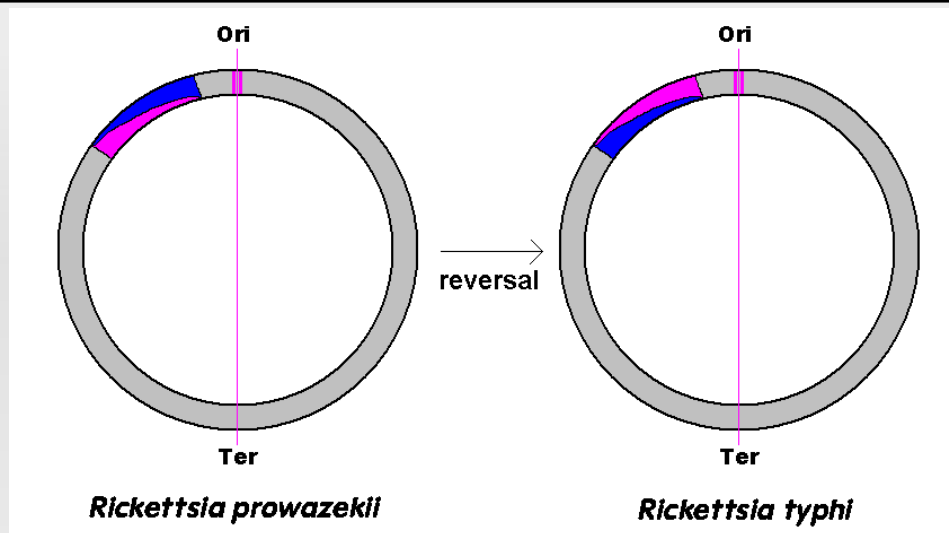
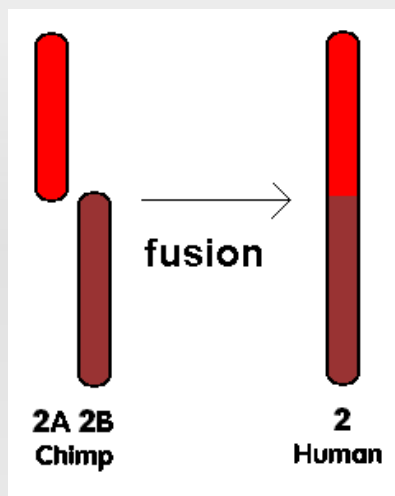
# Introduction

## Genome rearrangements



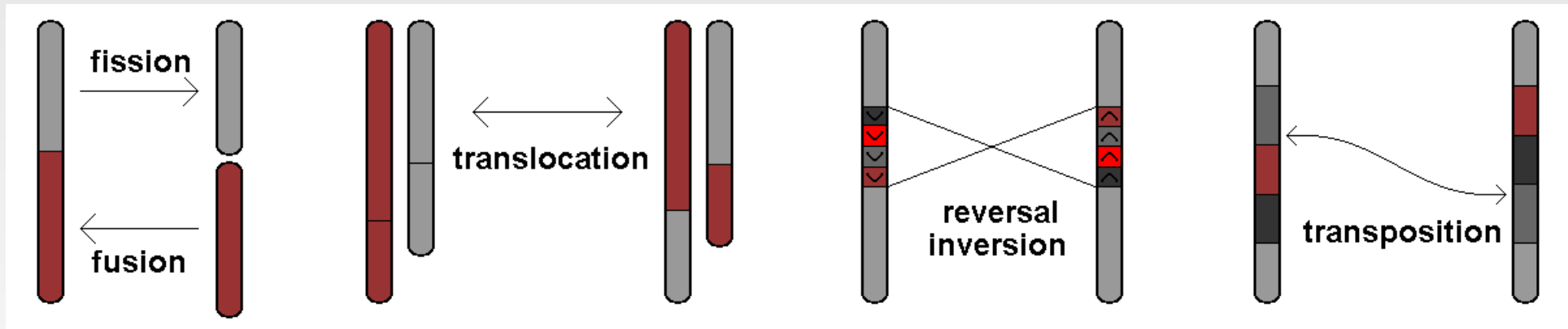
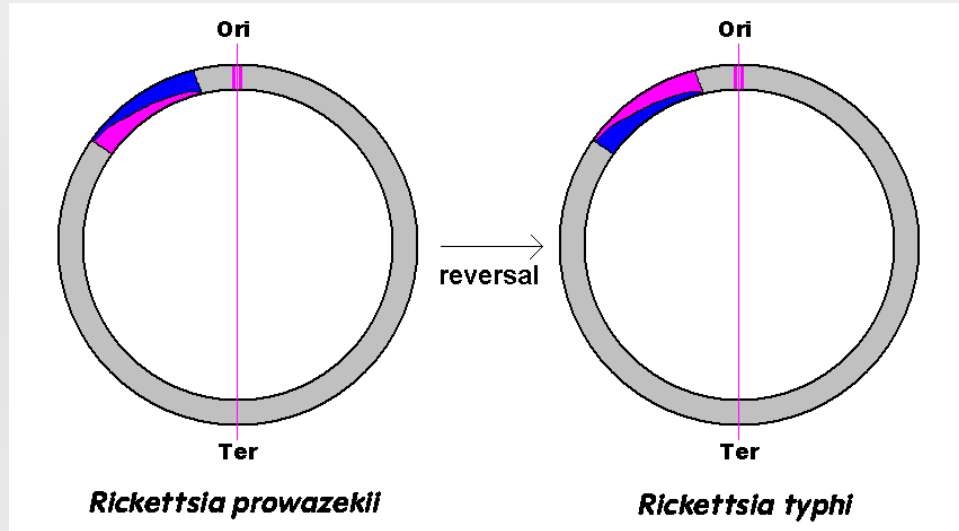
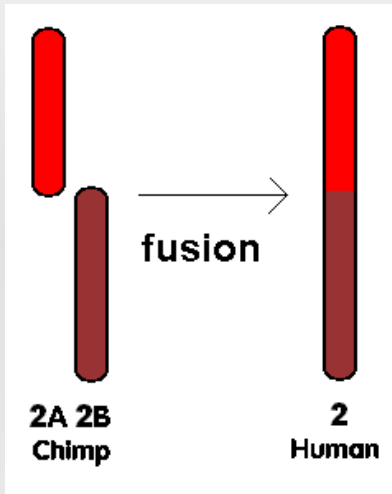
# Introduction

## Genome rearrangements



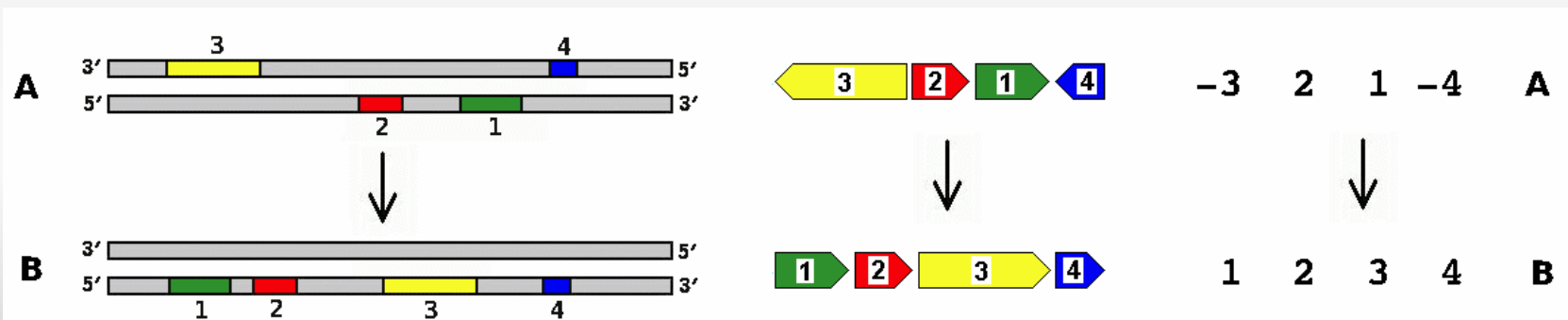
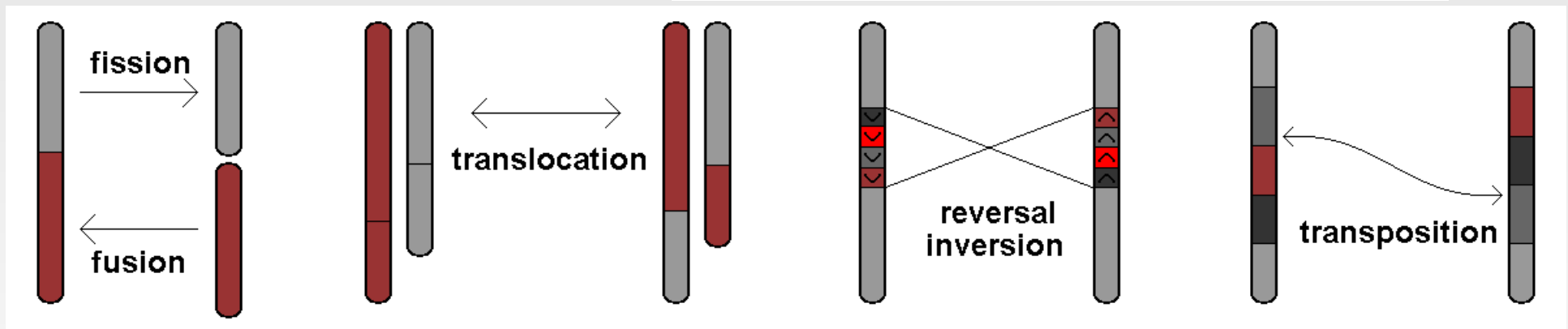
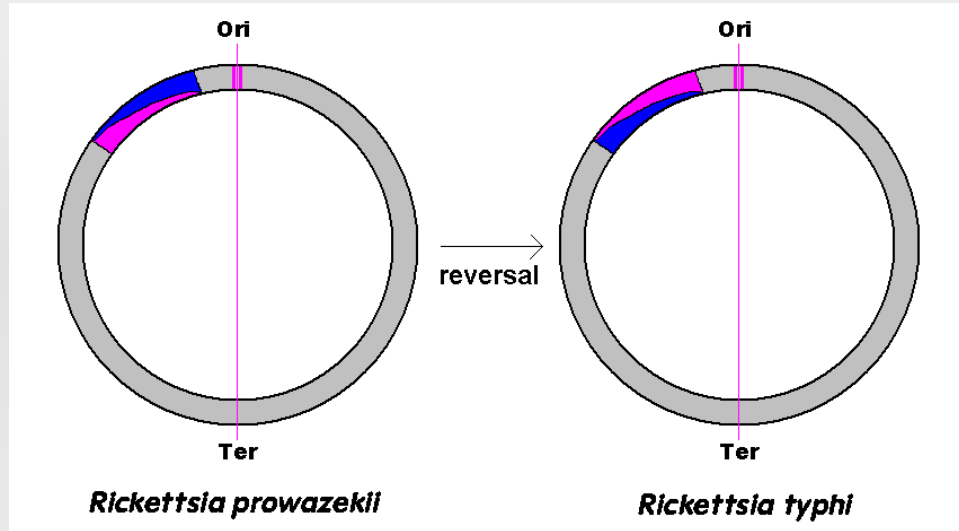
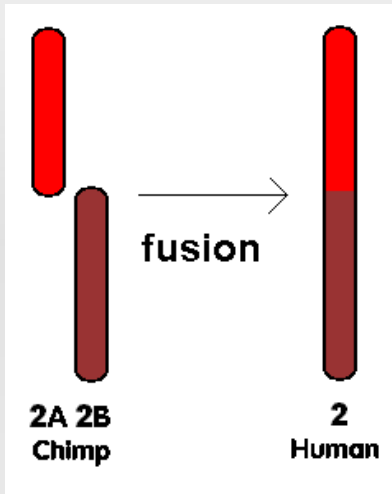
# Introduction

## Genome rearrangements



# Introduction

## Genome rearrangements



## Sorting by reversals (SR)

## Sorting by reversals (SR)

Only reversals are considered : genomes are unichromosomal

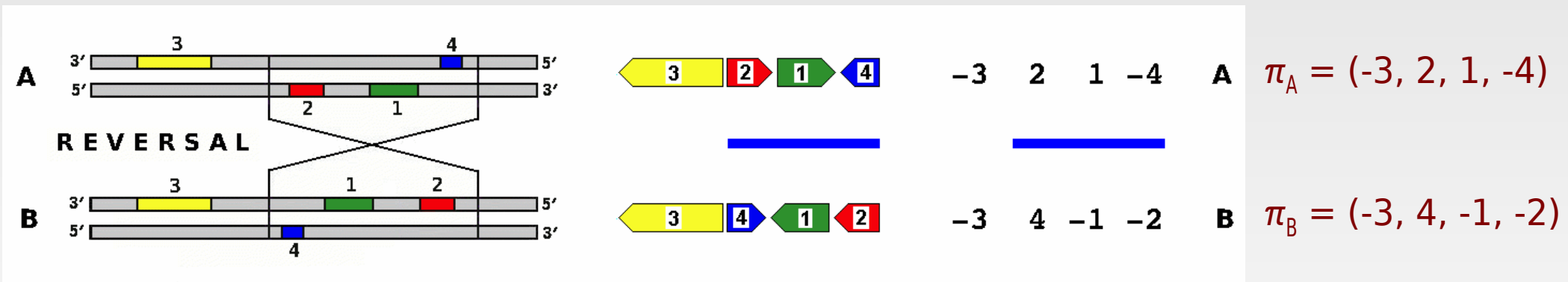
Duplications are not allowed

# Sorting by reversals (SR)

Only reversals are considered : genomes are unichromosomal

Duplications are not allowed

reversal  $\Rightarrow \rho = \{1, 2, 4\}$

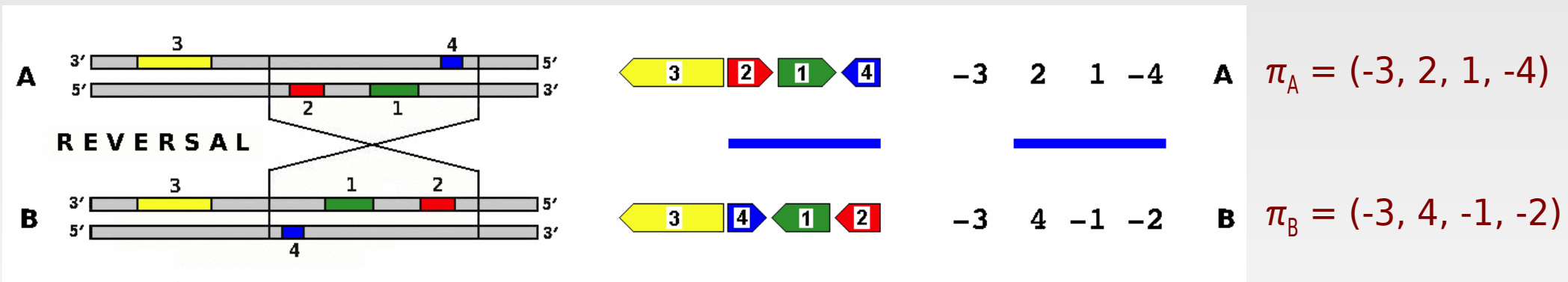


# Sorting by reversals (SR)

Only reversals are considered : genomes are unichromosomal

Duplications are not allowed

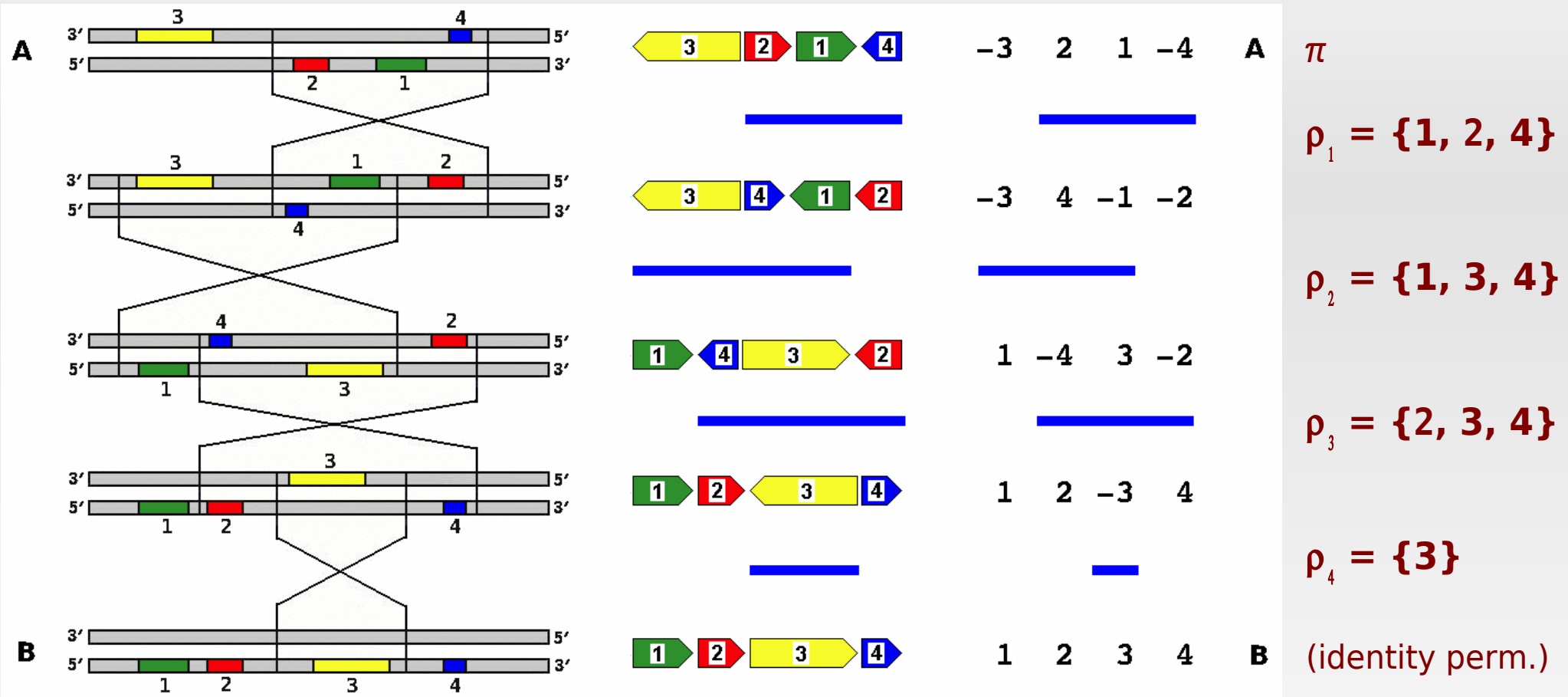
reversal  $\Rightarrow \rho = \{1, 2, 4\}$



Two classical problems:

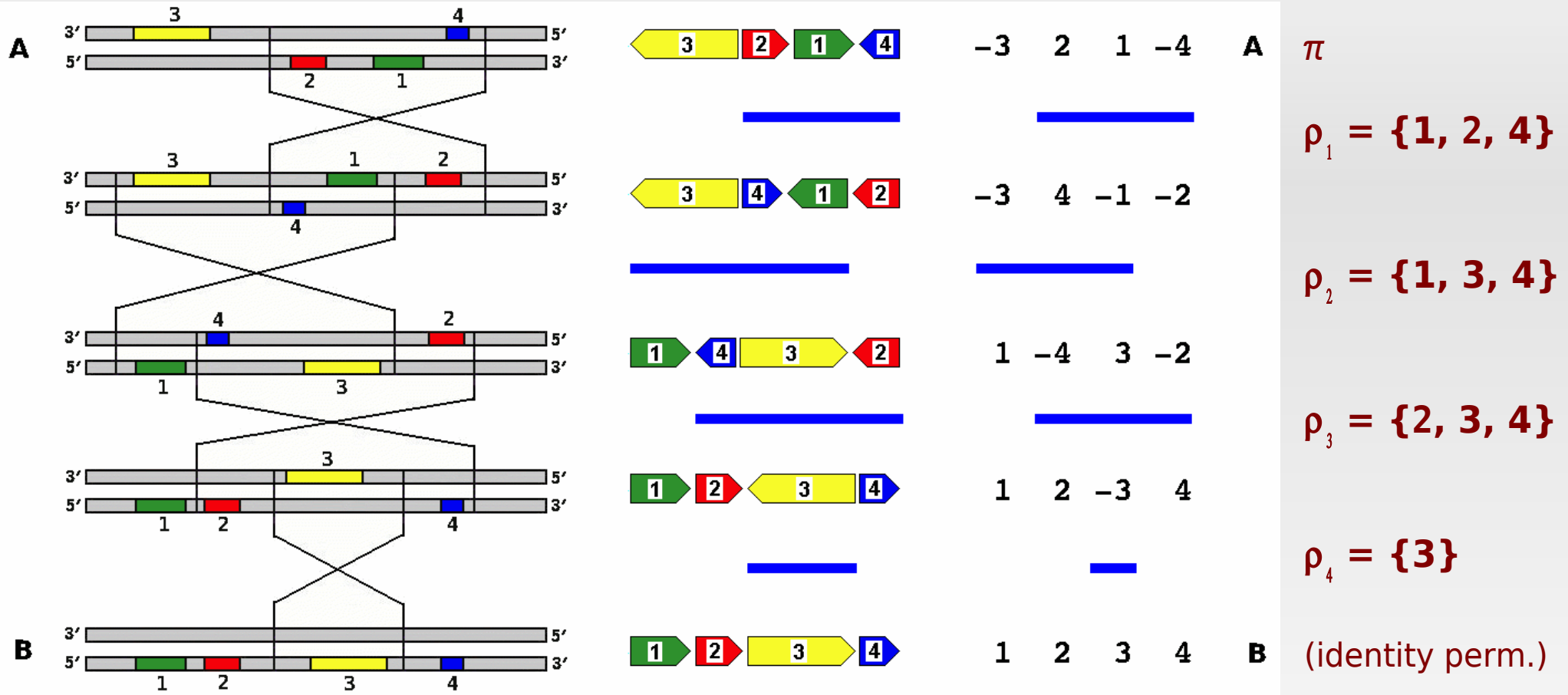
- Computing the **reversal distance**  $d(\pi)$
- Finding one **optimal sorting sequence**

# Sorting by reversals (SR)



The sequence  $\{1, 2, 4\} \{1, 3, 4\} \{2, 3, 4\} \{3\}$  is an **optimal sorting sequence**

# Sorting by reversals (SR)



The sequence  $\{1, 2, 4\} \{1, 3, 4\} \{2, 3, 4\} \{3\}$  is an **optimal sorting sequence**

(This approach is symmetric)

## Sorting by reversals (SR)

- Given a permutation  $\pi$ , **calculating the reversal distance** and **finding one optimal sorting sequence for  $\pi$**  can be computed in **polynomial time** (Hannenhalli and Pevzner, 1995)
  - . Calculating the reversal distance is  $O(n)$  (Bader et al., 2000)
  - . Finding one optimal solution is  $< O(n^2)$  (Tannier et al., 2007)
- Main available tools: GRIMM and GRAPPA
- Several approaches find **one** (arbitrary) optimal sorting sequence, while there might be several different optimal sequences

# The solution space of SR

Siepel (2003) proposed an algorithm that gives all optimal **next reversals** for a given permutation  $\pi$

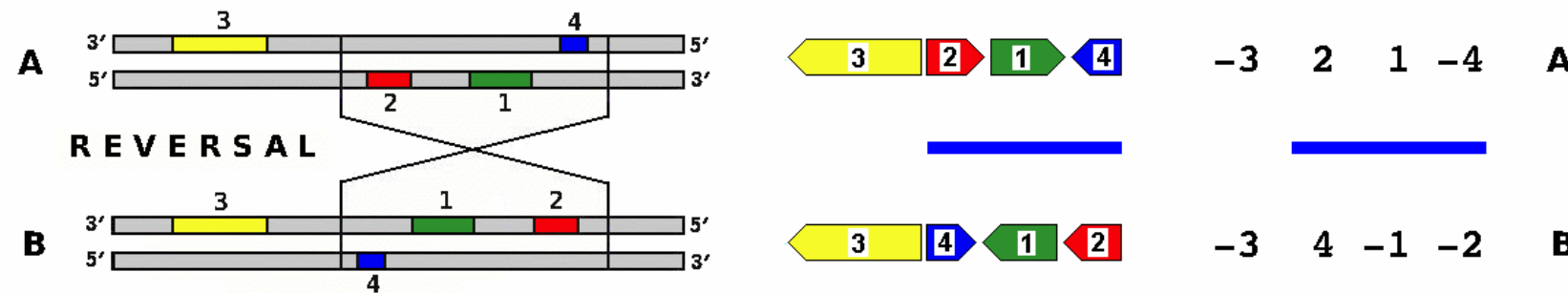
( the algorithm runs in  $O(n^3)$ ,  
the number of returned reversals is  $O(n^2)$  )

# The solution space of SR

Siepel (2003) proposed an algorithm that gives all optimal **next reversals** for a given permutation  $\pi$

( the algorithm runs in  $O(n^3)$ ,  
the number of returned reversals is  $O(n^2)$  )

Next reversals:



{1}, {1,2,3}, {2},  
{3}, {1,2,4}, {4}

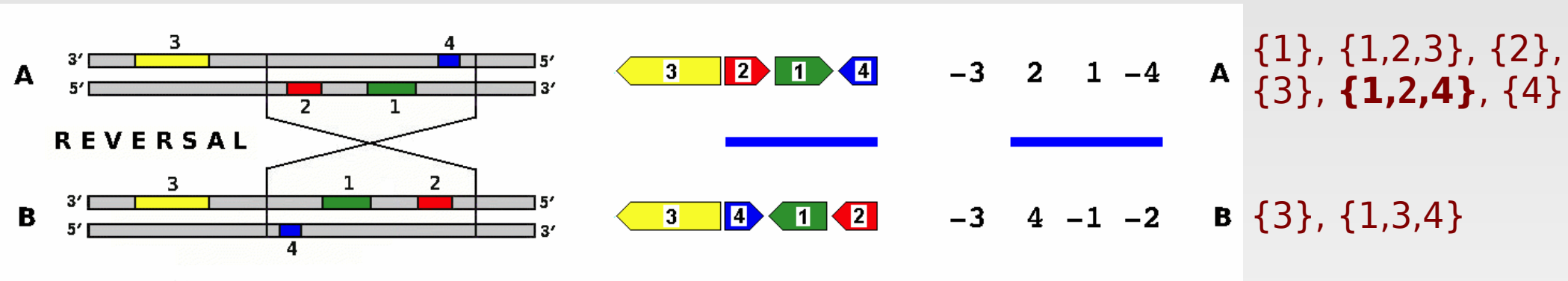
{3}, {1,3,4}

# The solution space of SR

Siepel (2003) proposed an algorithm that gives all optimal **next reversals** for a given permutation  $\pi$

( the algorithm runs in  $O(n^3)$ ,  
the number of returned reversals is  $O(n^2)$  )

Next reversals:



This algorithm allows the **enumeration of all existing optimal sorting sequences** for  $\pi$ .

(but the **number of optimal sorting sequences** is usually **huge**)

# The solution space of SR

$$\pi = (-3, 2, 1, -4)$$

$$\mathbf{d} = 4 ; \mathbf{s} = 28$$

$$\pi = (-6, 5, 7, -1, -4, 3, 2)$$

$$\mathbf{d} = 6 ; \mathbf{s} = 496$$

$$\pi = (-4, -3, 12, -11, -8, 10, 9, 7, -6, -5, 2, -1)$$

$$\mathbf{d} = 8 ; \mathbf{s} = 31\ 752$$

$$\pi = (-4, 3, 12, -11, -8, 10, 9, 7, -6, -5, 2, -1)$$

$$\mathbf{d} = 9 ; \mathbf{s} = 407\ 232$$

$$\pi = (-12, 11, -10, 6, 13, -5, 2, 7, 8, -9, 3, 4, 1)$$

$$\mathbf{d} = 10 ; \mathbf{s} = 8\ 278\ 540$$

$$\pi = (-12, 11, -10, -1, 16, -4, -3, 15, -14, 9, -8, -7, -2, -13, 5, -6)$$

$$\mathbf{d} = 12 ; \mathbf{s} = 505\ 634\ 256$$

$$\pi = (-12, 11, -10, 6, -5, 13, 2, 7, 8, -9, 14, -15, 3, 4, -16, 1)$$

$$\mathbf{d} = 13 ; \mathbf{s} = 40\ 313\ 272\ 766$$

Bergeron et al (2002):

- Many optimal solutions are **equivalent**:

$$\pi = (-3, 2, 1, -4)$$

{1, 2, 4} {1, 3, 4} {2, 3, 4} {3}

{1, 2, 4} {1, 3, 4} {3} {2, 3, 4}

{1, 2, 4} {3} {1, 3, 4} {2, 3, 4}

{3} {1, 2, 4} {1, 3, 4} {2, 3, 4}

## Traces

Bergeron et al (2002):

- Many optimal solutions are **equivalent**:

$$\pi = (-3, 2, 1, -4)$$

{1, 2, 4} {1, 3, 4} {2, 3, 4} {3}  
{1, 2, 4} {1, 3, 4} {3} {2, 3, 4}  
{1, 2, 4} {3} {1, 3, 4} {2, 3, 4}  
{3} {1, 2, 4} {1, 3, 4} {2, 3, 4}

(A **trace** is a set of **optimal solutions** composed by the **same reversals** but in **different orders**)

# The solution space of SR

## Traces

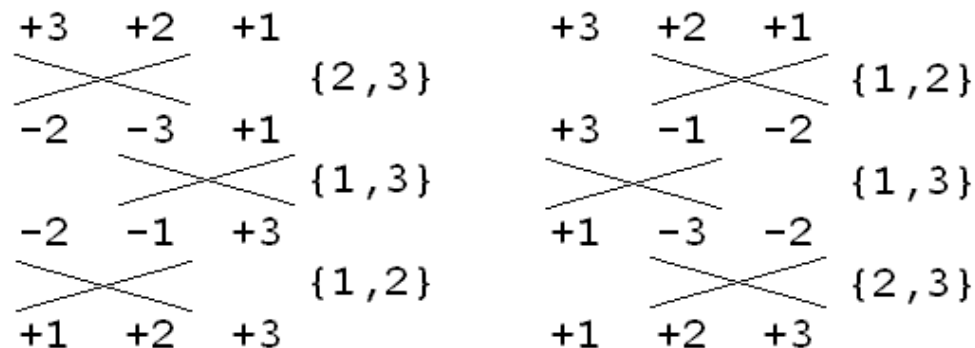
Bergeron et al (2002):

- Many optimal solutions are **equivalent**:

$$\pi = (-3, 2, 1, -4)$$

{1, 2, 4} {1, 3, 4} {2, 3, 4} {3}  
{1, 2, 4} {1, 3, 4} {3} {2, 3, 4}  
{1, 2, 4} {3} {1, 3, 4} {2, 3, 4}  
{3} {1, 2, 4} {1, 3, 4} {2, 3, 4}

(A **trace** is a set of **optimal solutions** composed by the **same reversals** but in **different orders**)



The trace {2,3}{1,3}{1,2} is different from the trace {1,2}{1,3}{2,3}

But not every pair of solutions composed by the same reversals are in the same trace...

## Traces

Two reversals  $\rho$  and  $\theta$  **commute** if they are **disjoint sets** or if **one is a subset of the other**.

$\{1,3,4\}$  and  $\{2,5\}$  commute

$\{1,3,4\}$  and  $\{3\}$  commute

$\{1,2,4\}$  and  $\{1,3,4\}$  do not commute

## Traces

Two reversals  $\rho$  and  $\theta$  **commute** if they are **disjoint sets** or if **one is a subset of the other**.

$\{1,3,4\}$  and  $\{2,5\}$  commute

$\{1,3,4\}$  and  $\{3\}$  commute

$\{1,2,4\}$  and  $\{1,3,4\}$  do not commute

$\rho$  and  $\theta$  commute :  $\dots\rho\theta\dots$  is equivalent to  $\dots\theta\rho\dots$

$\{1, 2, 4\} \{1, 3, 4\} \{3\} \{2, 3, 4\}$  is equivalent to  
 $\{1, 2, 4\} \{3\} \{1, 3, 4\} \{2, 3, 4\}$

(but  $\{2,3\}\{1,3\}\{1,2\}$  is not equivalent to  $\{1,2\}\{1,3\}\{2,3\}$ )

## Traces

Two reversals  $\rho$  and  $\theta$  **commute** if they are **disjoint sets** or if **one is a subset of the other**.

$\{1,3,4\}$  and  $\{2,5\}$  commute  
 $\{1,3,4\}$  and  $\{3\}$  commute  
 $\{1,2,4\}$  and  $\{1,3,4\}$  do not commute

$\rho$  and  $\theta$  commute :  $\dots\rho\theta\dots$  is equivalent to  $\dots\theta\rho\dots$

$\{1, 2, 4\} \{1, 3, 4\} \{3\} \{2, 3, 4\}$  is equivalent to  
 $\{1, 2, 4\} \{3\} \{1, 3, 4\} \{2, 3, 4\}$

(but  $\{2,3\}\{1,3\}\{1,2\}$  is not equivalent to  $\{1,2\}\{1,3\}\{2,3\}$ )

A **trace** is a set of **optimal sequences** which are all **equivalent** under the transitive closure of this **commuting relation**.

## Traces

The traces are a compact representation of the **set of all optimal sorting sequences** for a permutation  $\pi$

**Finding** an element of **each trace without enumerating all solutions** was stated to be an **open problem**

M. Braga, M.-F. Sagot, C. Scornavacca and E. Tannier (2007): **an algorithm to enumerate all the traces and to give the number of elements in each trace**

$$\pi = (-3, 2, 1, -4)$$

{1, 2, 4} {1, 3, 4} {2, 3, 4} {3} (4 solutions)

{1} {1, 2, 3} {2} {4} (24 solutions)

# The solution space of SR

## Enumerating traces

**4-trace:** (each element has 4 reversals)

{1, 2, 4} {1, 3, 4} {2, 3, 4} {3}  
{1, 2, 4} {1, 3, 4} {3} {2, 3, 4}  
{1, 2, 4} {3} {1, 3, 4} {2, 3, 4}  
{3} {1, 2, 4} {1, 3, 4} {2, 3, 4}

# The solution space of SR

## Enumerating traces

### 4-trace:

{1, 2, 4} {1, 3, 4} {2, 3, 4} {3}  
{1, 2, 4} {1, 3, 4} {3} {2, 3, 4}  
{1, 2, 4} {3} {1, 3, 4} {2, 3, 4}  
{3} {1, 2, 4} {1, 3, 4} {2, 3, 4}

### Prefixes of a trace:

# The solution space of SR

## Enumerating traces

### 4-trace:

{1, 2, 4} {1, 3, 4} {2, 3, 4} {3}  
{1, 2, 4} {1, 3, 4} {3} {2, 3, 4}  
{1, 2, 4} {3} {1, 3, 4} {2, 3, 4}  
{3} {1, 2, 4} {1, 3, 4} {2, 3, 4}

### Prefixes of a trace:

### 1-prefixes:

{1, 2, 4}

{3}

# The solution space of SR

## Enumerating traces

### 4-trace:

{1, 2, 4} {1, 3, 4} {2, 3, 4} {3}  
{1, 2, 4} {1, 3, 4} {3} {2, 3, 4}  
{1, 2, 4} {3} {1, 3, 4} {2, 3, 4}  
{3} {1, 2, 4} {1, 3, 4} {2, 3, 4}

### Prefixes of a trace:

#### 2-prefixes:

{1, 2, 4} {1, 3, 4}

{1, 2, 4} {3}  
{3} {1, 2, 4}

#### 1-prefixes:

{1, 2, 4}

{3}

## Enumerating traces

### 4-trace:

{1, 2, 4} {1, 3, 4} {2, 3, 4} {3}  
{1, 2, 4} {1, 3, 4} {3} {2, 3, 4}  
{1, 2, 4} {3} {1, 3, 4} {2, 3, 4}  
{3} {1, 2, 4} {1, 3, 4} {2, 3, 4}

### 3-prefixes:

{1, 2, 4} {1, 3, 4} {2, 3, 4} (a **3-trace** with only **one** element)

{1, 2, 4} {1, 3, 4} {3}  
{1, 2, 4} {3} {1, 3, 4} (a **3-trace** with **three** elements)  
{3} {1, 2, 4} {1, 3, 4}

### Prefixes of a trace:

### 2-prefixes:

{1, 2, 4} {1, 3, 4}

{1, 2, 4} {3}  
{3} {1, 2, 4}

### 1-prefixes:

{1, 2, 4}

{3}

# The solution space of SR

## Enumerating traces

### 4-trace:

{1, 2, 4} {1, 3, 4} {2, 3, 4} {3}  
{1, 2, 4} {1, 3, 4} {3} {2, 3, 4}  
{1, 2, 4} {3} {1, 3, 4} {2, 3, 4}  
{3} {1, 2, 4} {1, 3, 4} {2, 3, 4}

### 3-prefixes:

{1, 2, 4} {1, 3, 4} {2, 3, 4} (a **3-trace** with  
only **one** element)

{1, 2, 4} {1, 3, 4} {3}  
{1, 2, 4} {3} {1, 3, 4} (a **3-trace** with  
{3} {1, 2, 4} {1, 3, 4} **three** elements)

### Prefixes of a trace:

### 2-prefixes:

{1, 2, 4} {1, 3, 4}

{1, 2, 4} {3}  
{3} {1, 2, 4}

### 1-prefixes:

{1, 2, 4}

{3}

[ The **size** of the  
**4-trace** is the  
**sum of the sizes**  
of its **3-prefixes** ]

The solution space of SR

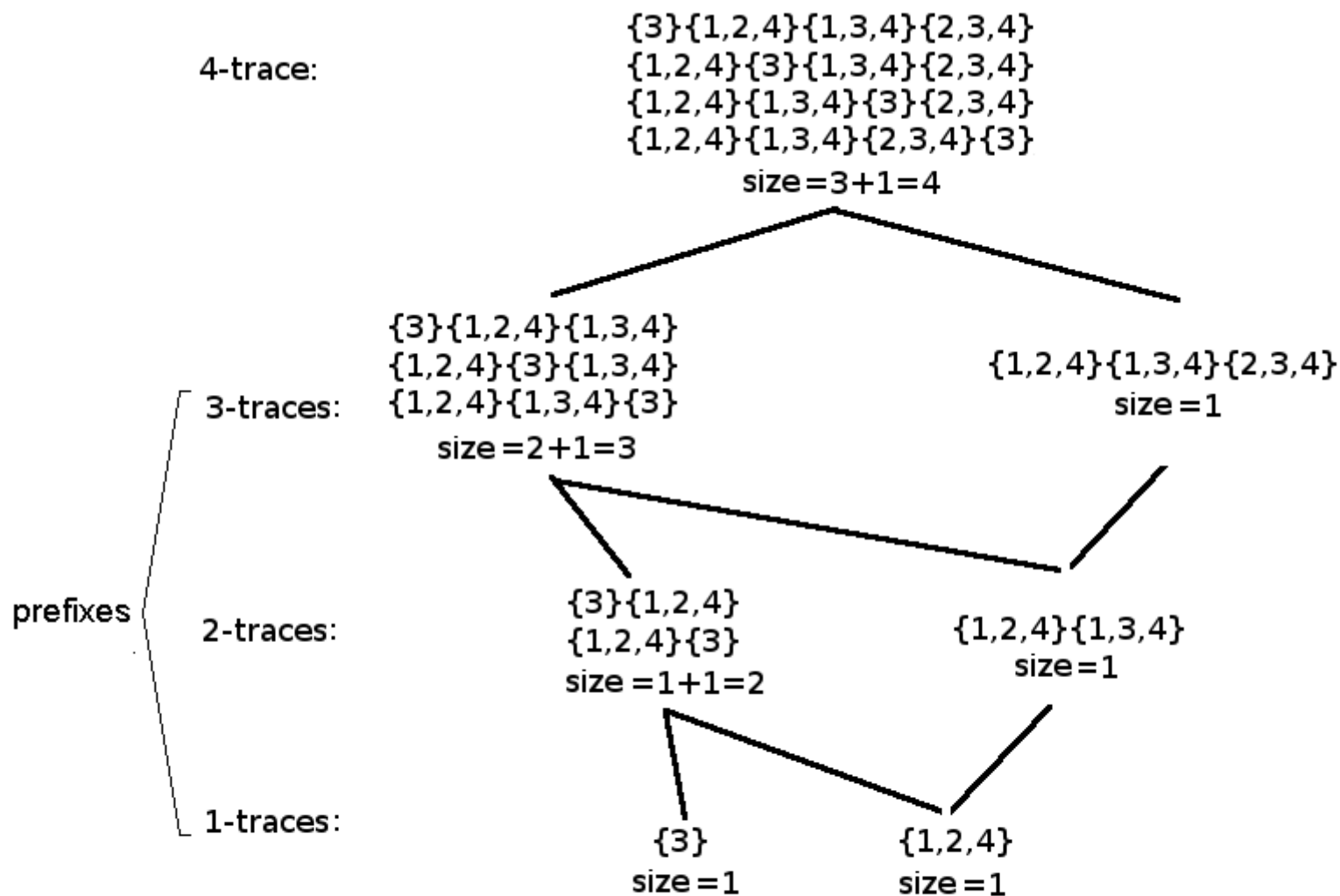
## Enumerating traces

The **size** (number of elements) of an **i-trace** is the **sum of the sizes** of its **(i-1)-prefixes**.

# The solution space of SR

## Enumerating traces

The **size** (number of elements) of an **i-trace** is the **sum of the sizes** of its **(i-1)-prefixes**.

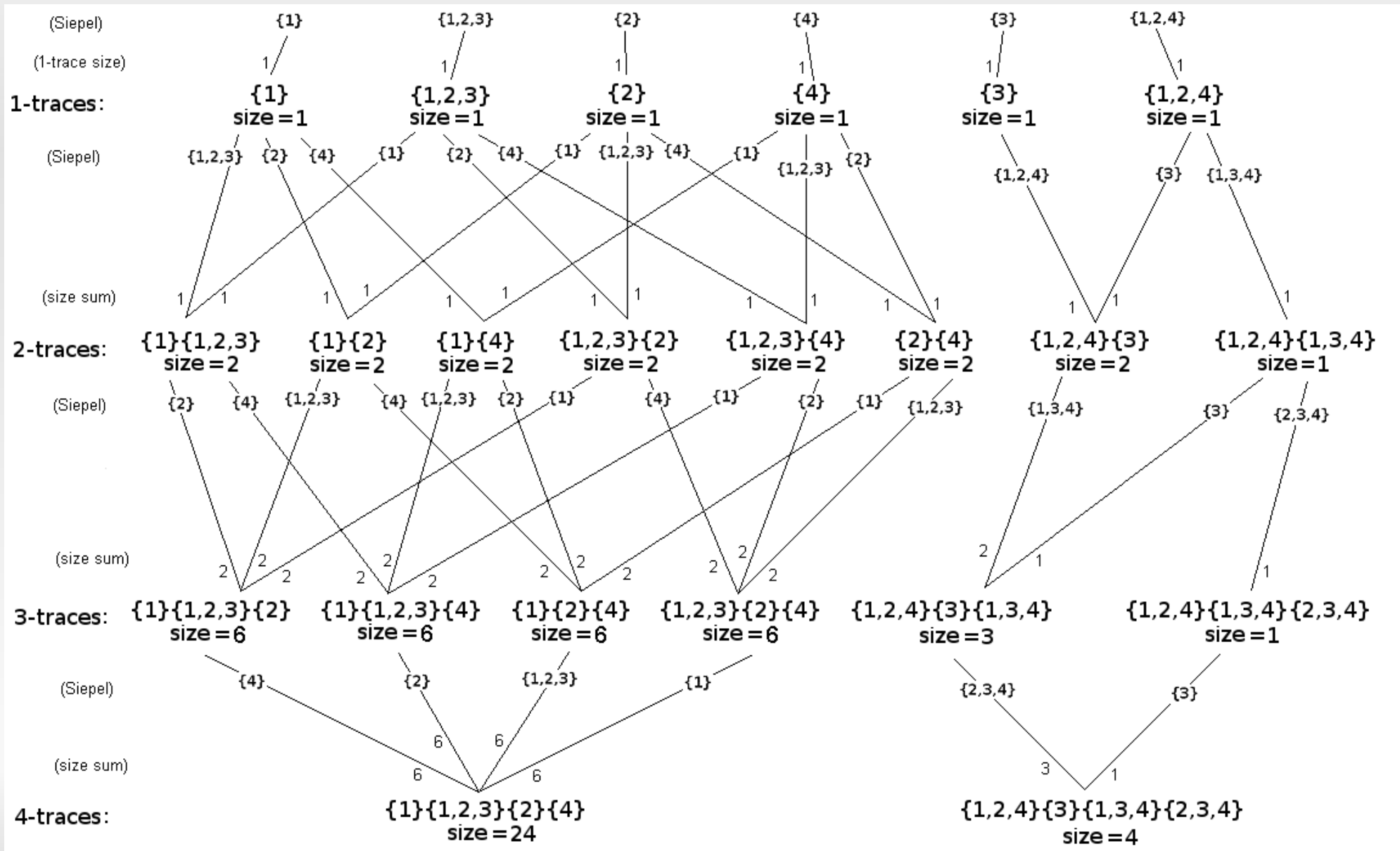


The **size** of a **1-trace** is **1** (trivial)

# The solution space of SR

## Enumerating traces

$$[\pi = (-3, 2, 1, -4)]$$



The solution space of SR

Enumerating traces [ complexity ]

Theoretical complexity:  $O( N \cdot n^{k_{\max} + 4} )$

The solution space of SR

Enumerating traces [ complexity ]

**Theoretical complexity:  $O( N \cdot n^{k_{\max} + 4} )$**

The **width (k)** of a trace  $t$  is defined as the **biggest subset** of reversals of  $t$  such that **every pair of reversals** in this subset **commutes**.

# The solution space of SR

## Enumerating traces [ complexity ]

**Theoretical complexity:  $O( N \cdot n^{k_{\max} + 4} )$**

The **width (k)** of a trace  $t$  is defined as the **biggest subset** of reversals of  $t$  such that **every pair of reversals** in this subset **commutes**.

$\{1, 2, 4\} \{1, 3, 4\} \{2, 3, 4\} \{3\}$

Subsets:

$\{ \{1, 2, 4\}, \{3\} \}$ , size = 2

$\{ \{1, 3, 4\}, \{3\} \}$ , size = 2

$\{ \{2, 3, 4\}, \{3\} \}$ , size = 2

$k = 2$

# The solution space of SR

## Enumerating traces [ complexity ]

**Theoretical complexity:  $O( N \cdot n^{k_{\max} + 4} )$**

The **width (k)** of a trace  $t$  is defined as the **biggest subset** of reversals of  $t$  such that **every pair of reversals** in this subset **commutes**.

$\{1, 2, 4\}\{1, 3, 4\}\{2, 3, 4\}\{3\}$

Subsets:

$\{ \{1, 2, 4\}, \{3\} \}$ , size = 2

$\{ \{1, 3, 4\}, \{3\} \}$ , size = 2

$\{ \{2, 3, 4\}, \{3\} \}$ , size = 2

$k = 2$

$\{1\}\{1, 2, 3\}\{2\}\{4\}$

Subsets:

$\{ \{1\}, \{1, 2, 3\}, \{2\}, \{4\} \}$ , size = 4

$k = 4$

# The solution space of SR

## Enumerating traces [ complexity ]

**Theoretical complexity:  $O( N \cdot n^{k_{\max} + 4} )$**

The **width (k)** of a trace  $t$  is defined as the **biggest subset** of reversals of  $t$  such that **every pair of reversals** in this subset **commutes**.

$\{1, 2, 4\} \{1, 3, 4\} \{2, 3, 4\} \{3\}$

Subsets:

$\{ \{1, 2, 4\}, \{3\} \}$ , size = 2

$\{ \{1, 3, 4\}, \{3\} \}$ , size = 2

$\{ \{2, 3, 4\}, \{3\} \}$ , size = 2

$k = 2$

$\{1\} \{1, 2, 3\} \{2\} \{4\}$

Subsets:

$\{ \{1\}, \{1, 2, 3\}, \{2\}, \{4\} \}$ , size = 4

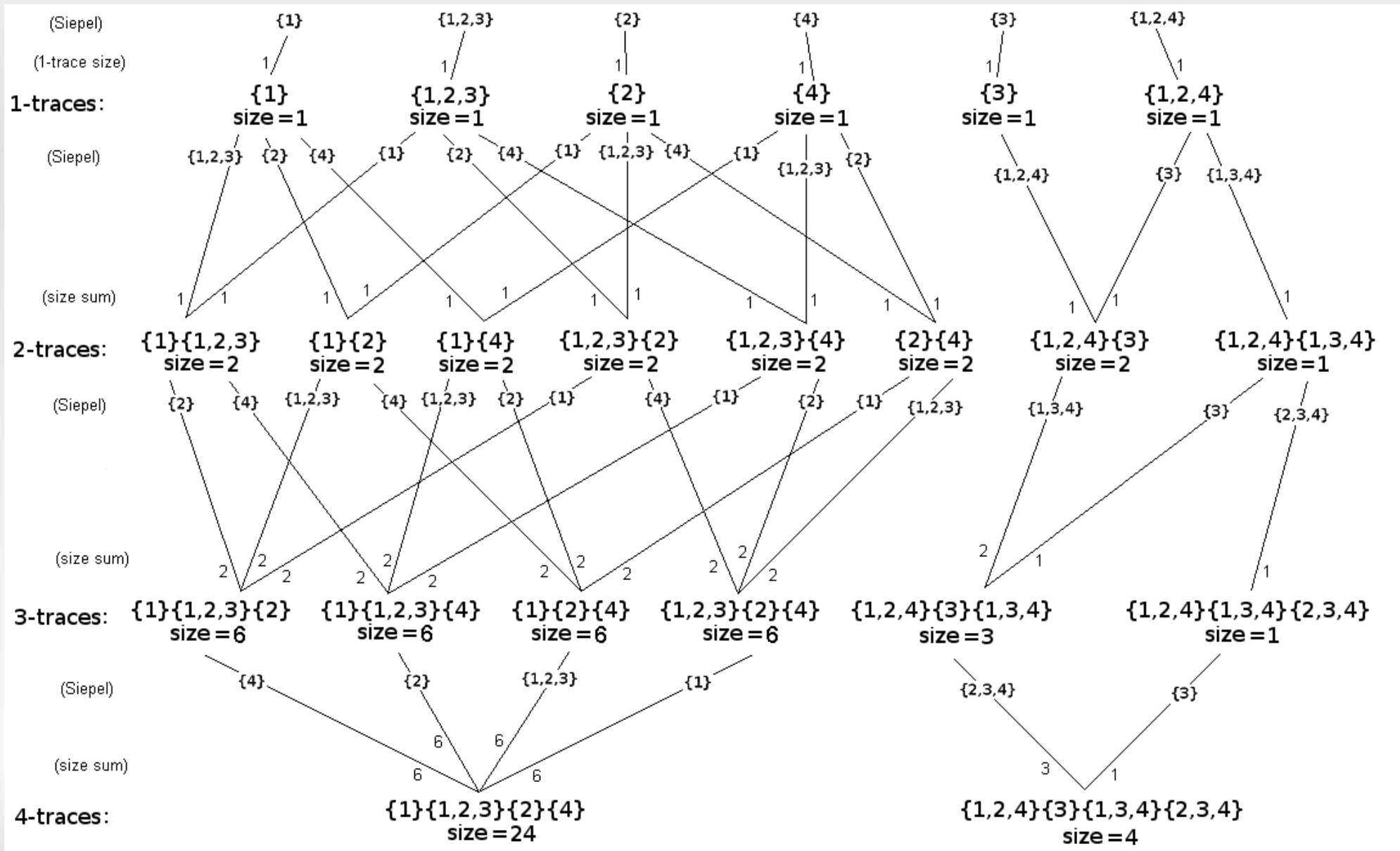
$k = 4$

**$k_{\max} = 4$**

For  $\pi = (-3, 2, 1, -4)$ , we have  $n = 4$ ,  $N = 2$  and  $k_{\max} = 4$

# The solution space of SR

## Enumerating traces [ $\pi = (-3, 2, 1, -4)$ ]



# The solution space of SR

## Enumerating traces [ experiments ]

The number of traces is much smaller than the number of sorting sequences

$$\pi_F = (-12, 11, -10, 6, 13, -5, 2, 7, 8, -9, 3, 4, 1)$$

$$\pi_G = (-12, 11, -10, -1, 16, -4, -3, 15, -14, 9, -8, -7, -2, -13, 5, -6)$$

$$\pi_H = (-12, 11, -10, 6, -5, 13, 2, 7, 8, -9, 14, -15, 3, 4, -16, 1)$$

PERMUT.	$N_S$	$N_T$	Algorithm	Execution time
$\pi_F$ $n = 12$ $d = 10$	8,278,540	2,151	enum seq. enum+traces traces	$\simeq 13.5$ min $\simeq 30.1$ min $\simeq 27$ sec
$\pi_G$ $n = 16$ $d = 12$	505,634,256	21,902	enum seq. enum+traces traces	$\simeq 16$ h $\simeq 43.5$ h $\simeq 7.3$ min
$\pi_H$ $n = 16$ $d = 13$	40,313,272,766	567,524	enum seq. enum+traces traces	- - $\simeq 4.1$ hours

# The solution space of SR

## Enumerating traces [ experiments ]

The number of traces is much smaller than the number of sorting sequences

$$\pi_F = (-12, 11, -10, 6, 13, -5, 2, 7, 8, -9, 3, 4, 1)$$

$$\pi_G = (-12, 11, -10, -1, 16, -4, -3, 15, -14, 9, -8, -7, -2, -13, 5, -6)$$

$$\pi_H = (-12, 11, -10, 6, -5, 13, 2, 7, 8, -9, 14, -15, 3, 4, -16, 1)$$

PERMUT.	$N_S$	$N_T$	Algorithm	Execution time
$\pi_F$ $n = 12$ $d = 10$	8,278,540	2,151	enum seq. enum+traces traces	$\simeq 13.5$ min $\simeq 30.1$ min $\simeq 27$ sec
$\pi_G$ $n = 16$ $d = 12$	505,634,256	21,902	enum seq. enum+traces traces	$\simeq 16$ h $\simeq 43.5$ h $\simeq 7.3$ min
$\pi_H$ $n = 16$ $d = 13$	40,313,272,766	567,524	enum seq. enum+traces traces	- - $\simeq 4.1$ hours

The **solution space** represented by traces is generally **too big** for direct human interpretation

The solution space of SR

Enumerating traces [ remarks ]

The algorithm **gives a representation of all solutions** of sorting signed permutations by reversals, **without enumerating all solutions** (an alternative to approaches that give one sorting sequence)

## The solution space of SR

# Enumerating traces [ remarks ]

The algorithm **gives a representation of all solutions** of sorting signed permutations by reversals, **without enumerating all solutions** (an alternative to approaches that give one sorting sequence)

Although the **solution space is dramatically reduced** when dealing with traces, it **is still too big** for direct human interpretation.

## The solution space of SR

# Enumerating traces [ remarks ]

The algorithm **gives a representation of all solutions** of sorting signed permutations by reversals, **without enumerating all solutions** (an alternative to approaches that give one sorting sequence)

Although the **solution space is dramatically reduced** when dealing with traces, it **is still too big** for direct human interpretation.

An **implementation** of this algorithm is available **on-line**, integrated to the **baobabLuna** framework (in general, **we are still limited** to permutations with reversal distance bounded by  $\sim 20$ )

# Biological constraints & Applications

The solution space can be reduced with the use of biological constraints

(The constraints are used to filter reversals)

# Biological constraints & Applications

The solution space can be reduced with the use of biological constraints

(The constraints are used to filter reversals)

A positive selection may depend on the chronology of reversals:

the use of constraints can lead to asymmetric approaches

# Biological constraints & Applications

The solution space can be reduced with the use of biological constraints

(The constraints are used to filter reversals)

A positive selection may depend on the chronology of reversals:

the use of constraints can lead to asymmetric approaches

Constraints:

Common intervals

Replication terminus symmetry on evolution of *Rickettsia* bacterium

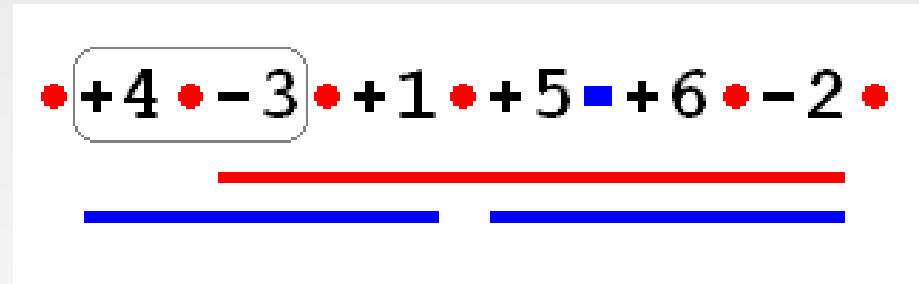
## Common intervals (CI)

**Common intervals** are intervals of the genomes composed by the same genes but not necessarily with the same order and orientations

## Common intervals (CI)

**Common intervals** are intervals of the genomes composed by the same genes but not necessarily with the same order and orientations

The genomes (4, -3, 1, 5, 6, -2)  
and (1, 2, 3, 4, 5, 6) have the following  
common intervals : {3,4} and {5,6}



# Biological constraints & Applications

## Common intervals (CI)

• -5 • -2 • -7 • +4 • -8 • +3 • +6 • -1 •

• -5 • -2 • +1 • -6 • -3 • +8 • -4 • +7 •

• -5 • -2 • +1 • +4 • -8 • +3 • +6 • +7 •

• -3 • +8 • -4 • -1 • +2 • +5 • +6 • +7 •

• -3 • -2 • +1 • +4 • -8 • +5 • +6 • +7 •

• -3 • -2 • -1 • +4 • -7 • -6 • -5 • +8 •

• -3 • -2 • -1 • +4 • +5 • +6 • +7 • +8 •

• +1 • +2 • +3 • +4 • +5 • +6 • +7 • +8 •

**Total:**

81869 solutions in  
**377** traces

# Biological constraints & Applications

## Common intervals (CI)

Selecting solutions that do not break common intervals:

•  $-5 \bullet -2 \bullet -7 \bullet +4 \bullet -8 \bullet +3 \bullet +6 \bullet -1 \bullet$   
—————  
•  $-5 \bullet -2 \bullet +1 \bullet -6 \bullet -3 \bullet +8 \bullet -4 \bullet +7 \bullet$   
—————  
•  $-5 \bullet -2 \bullet +1 \bullet +4 \bullet -8 \bullet +3 \bullet +6 \bullet +7 \bullet$   
—————  
•  $-3 \bullet +8 \bullet -4 \bullet -1 \bullet +2 \bullet +5 \bullet +6 \bullet +7 \bullet$   
—————  
•  $-3 \bullet -2 \bullet +1 \bullet +4 \bullet -8 \bullet +5 \bullet +6 \bullet +7 \bullet$   
— — — — —  
•  $-3 \bullet -2 \bullet -1 \bullet +4 \bullet -7 \bullet -6 \bullet -5 \bullet +8 \bullet$   
—————  
•  $-3 \bullet -2 \bullet -1 \bullet +4 \bullet +5 \bullet +6 \bullet +7 \bullet +8 \bullet$   
—————  
•  $+1 \bullet +2 \bullet +3 \bullet +4 \bullet +5 \bullet +6 \bullet +7 \bullet +8 \bullet$

•  $-5 \bullet -2 \bullet -7 \bullet +4 \bullet -8 \bullet +3 \bullet +6 \bullet -1 \bullet$   
—————  
•  $-3 \bullet +8 \bullet -4 \bullet +7 \bullet +2 \bullet +5 \bullet +6 \bullet -1 \bullet$   
—————  
•  $-8 \bullet +3 \bullet -4 \bullet +7 \bullet -2 \bullet +5 \bullet +6 \bullet -1 \bullet$   
—————  
•  $-8 \bullet +3 \bullet +2 \bullet -7 \bullet +4 \bullet +5 \bullet +6 \bullet -1 \bullet$   
—————  
•  $+1 \bullet -6 \bullet -5 \bullet -4 \bullet +7 \bullet -2 \bullet -3 \bullet +8 \bullet$   
—————  
•  $+1 \bullet +2 \bullet -7 \bullet +4 \bullet +5 \bullet +6 \bullet -3 \bullet +8 \bullet$   
—————  
•  $+1 \bullet +2 \bullet -7 \bullet -6 \bullet -5 \bullet -4 \bullet -3 \bullet +8 \bullet$   
—————  
•  $+1 \bullet +2 \bullet +3 \bullet +4 \bullet +5 \bullet +6 \bullet +7 \bullet +8 \bullet$

**Total:**

81869 solutions in  
**377** traces

**Common intervals:**

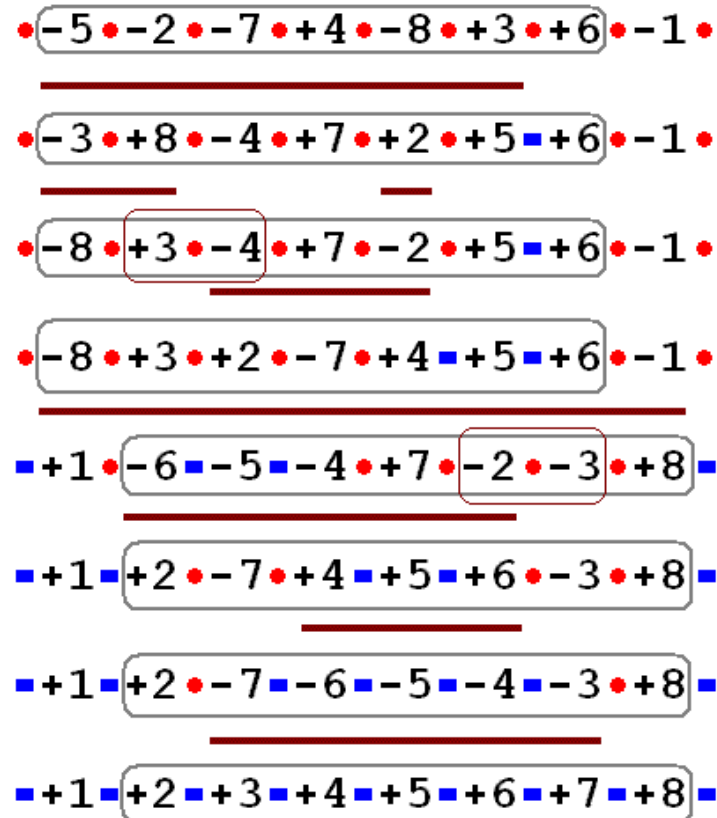
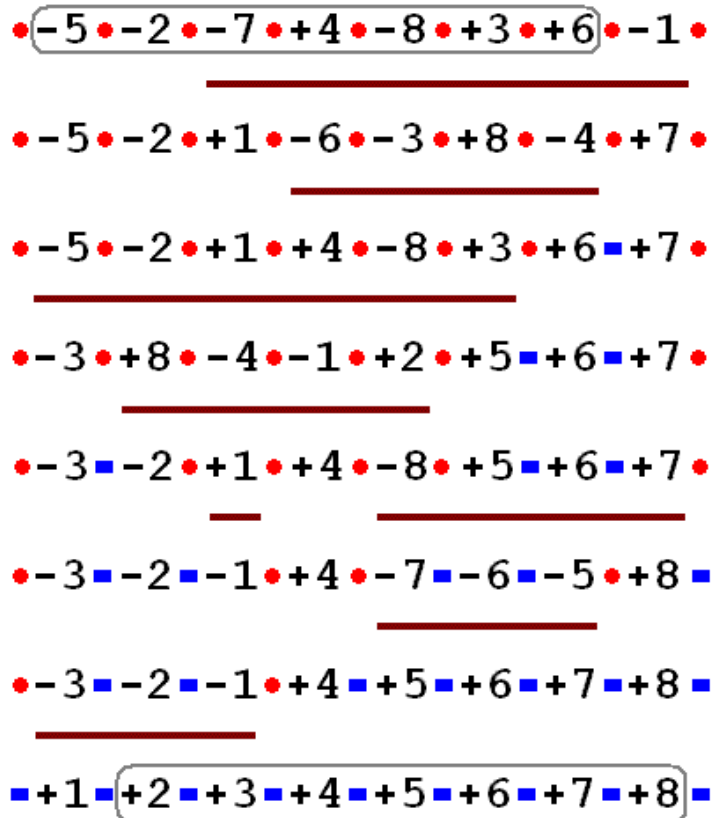
51304 solutions in  
**92** traces

(This approach  
is symmetric)

# Biological constraints & Applications

## Common intervals (CI)

Selecting solutions that do not break common intervals:



**Total:**

81869 solutions in  
**377** traces

**Common intervals:**

51304 solutions in  
**92** traces

(This approach  
is symmetric)

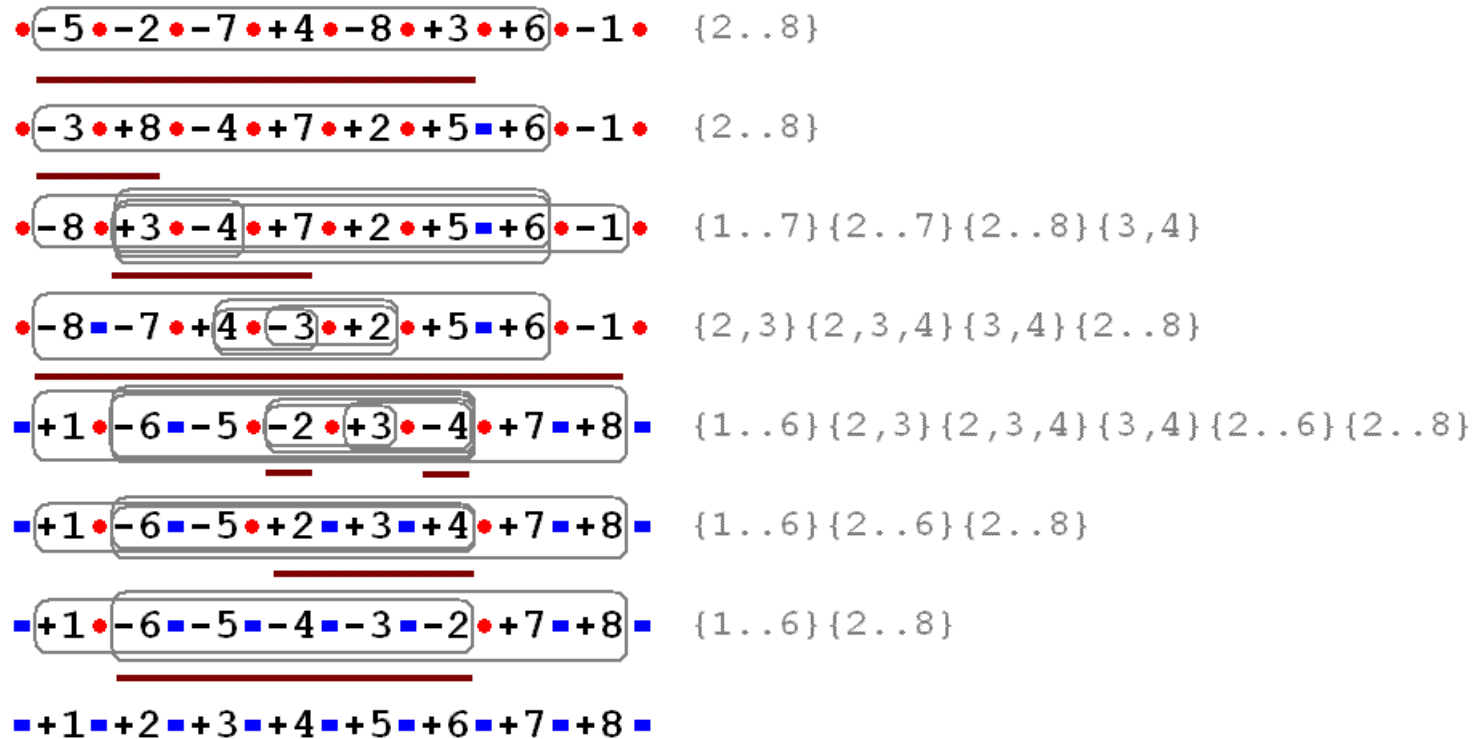
The new common intervals that appear in intermediary states are not considered...

# Biological constraints & Applications

## Progressive detection of CI

Selecting solutions that do not break  
**progressively** detected common intervals

Descendant



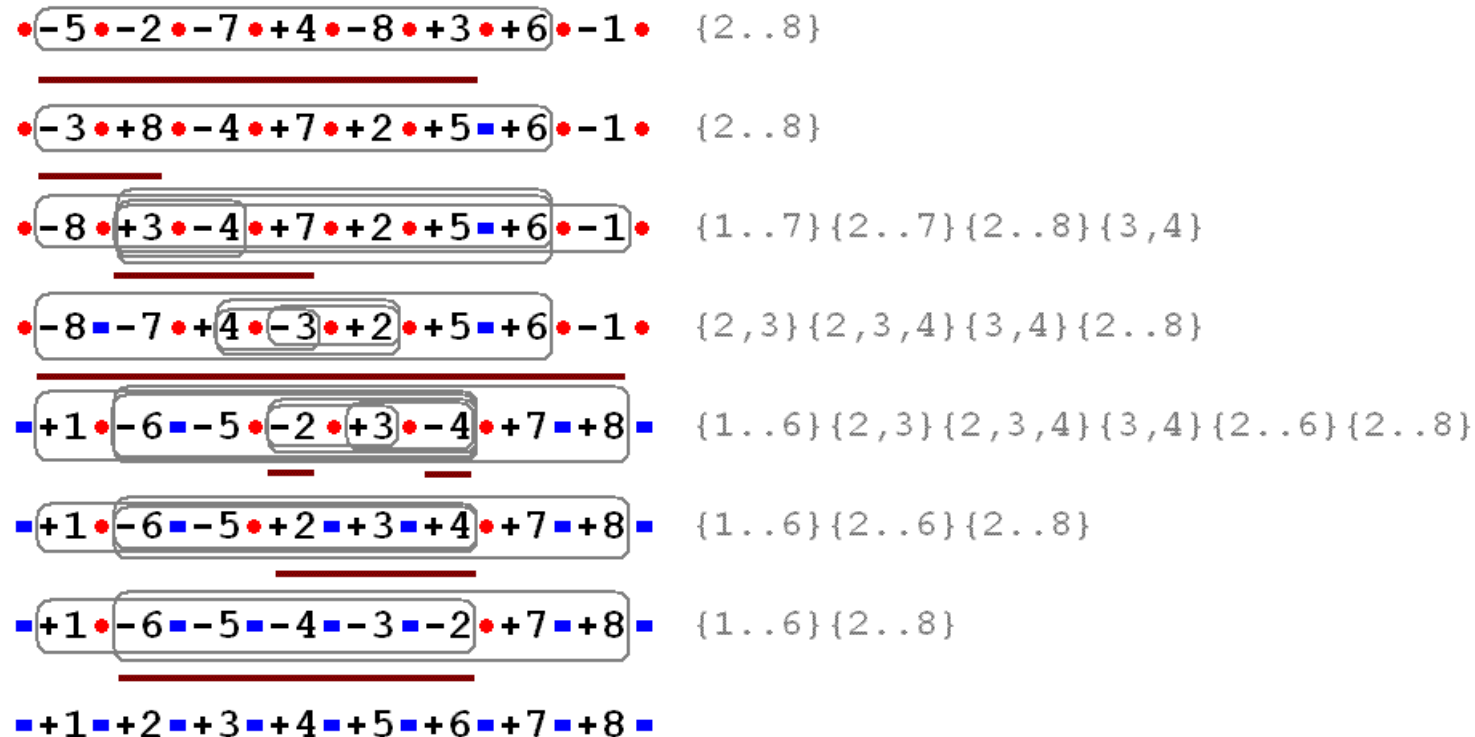
Ancestor

# Biological constraints & Applications

## Progressive detection of CI

Selecting solutions that do not break **progressively** detected common intervals

Descendant



Ancestor

(This approach is asymmetric)

# Biological constraints & Applications

## Progressive detection of CI

Selecting solutions that do not break **progressively** detected common intervals

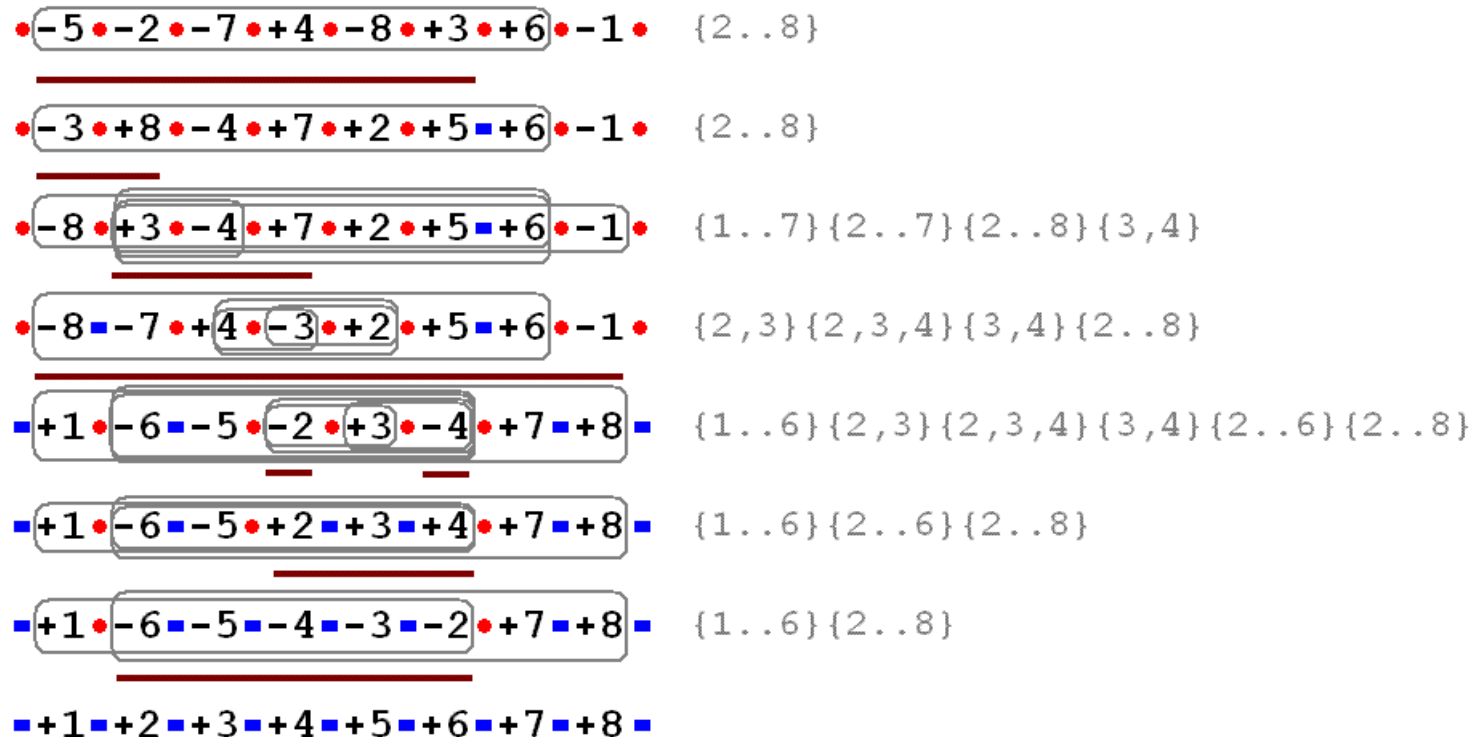
**Total:**

81869 solutions in  
**377** traces

**Common intervals:**

51304 solutions in  
**92** traces

**Descendant**



**Ancestor**

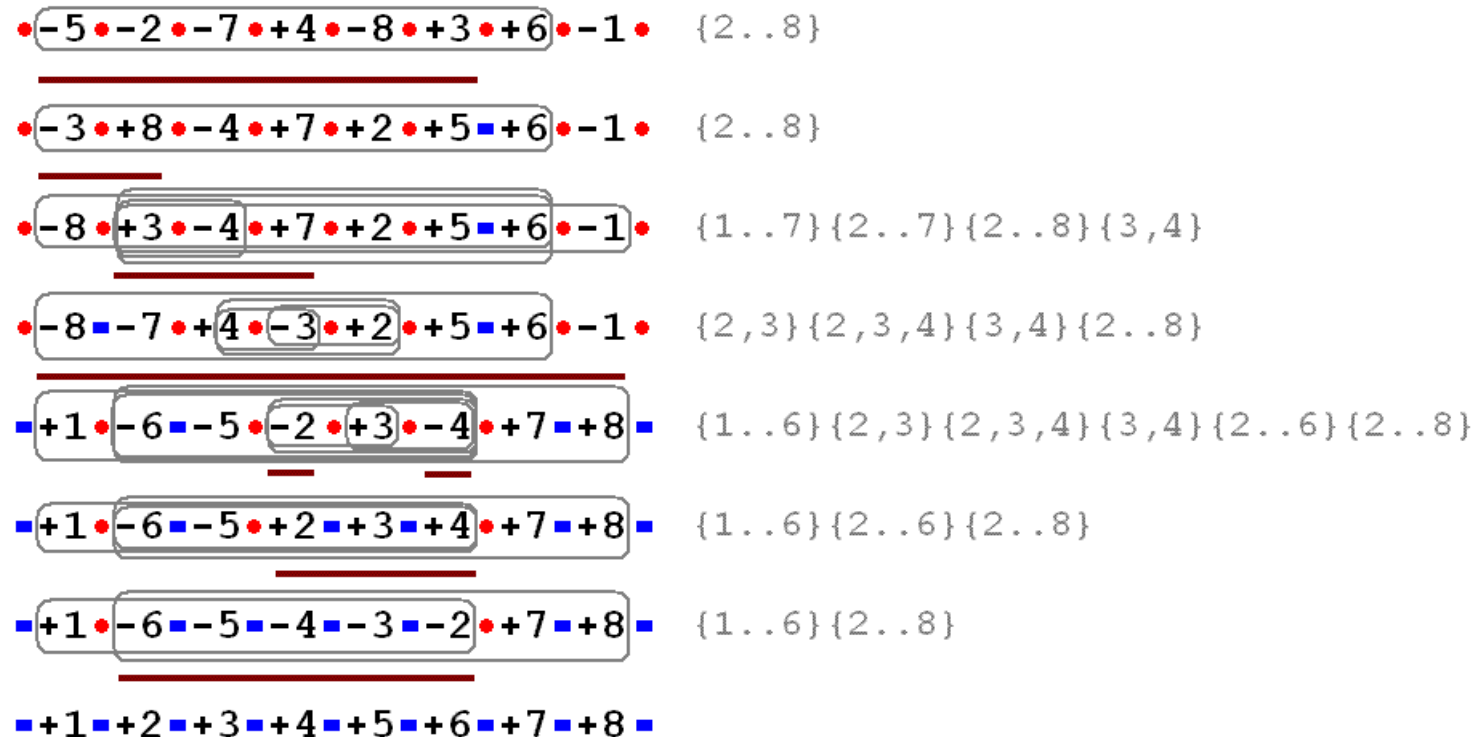
(This approach is  
asymmetric)

# Biological constraints & Applications

## Progressive detection of CI

Selecting solutions that do not break **progressively** detected common intervals

Descendant



Ancestor

(This approach is asymmetric)

**Total:**

81869 solutions in **377** traces

**Common intervals:**

51304 solutions in **92** traces

**Common intervals with progressive detection:**

11568 solutions in **12** subtraces from desc. to ancestor

and 8400 solutions in **5** subtraces from ancestor to desc.

# Biological constraints & Applications

## Common intervals (CI)

### Remarks:

The **complexity** of the algorithm does not change when we take common intervals into account with initial or progressive detection.

In practice, the **program runs faster** when we search for sequences that do not break the common intervals

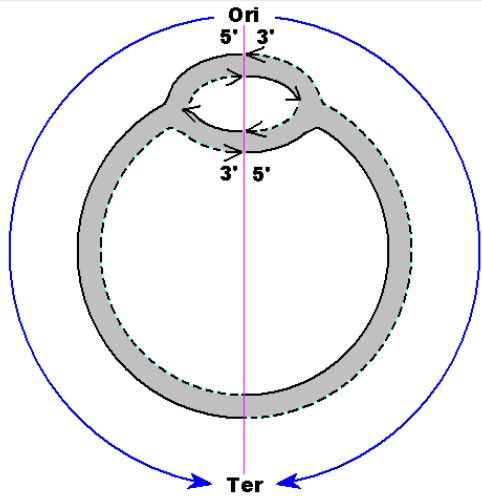
Permutation	Algorithm	$N_S$	$N_T$	Execution time
$B$ and $\mathcal{I}_{16}$ $d(B, \mathcal{I}_{16}) = 12$	all traces ( $B \leftrightarrow \mathcal{I}_{16}$ )	505,634,256	21,902	$\simeq 7.3$ minutes
	perfect traces ( $B \leftrightarrow \mathcal{I}_{16}$ )	122,862,960	171	$\simeq 27$ seconds
	p. perf. subtr. ( $B \rightarrow \mathcal{I}_{16}$ )	5,963,760	6	$\simeq 14$ seconds
	p. perf. subtr. ( $\mathcal{I}_{16} \rightarrow B$ )	5,393,520	9	$\simeq 16$ seconds

$B = (-12, 11, -10, -1, 16, -4, -3, 15, -14, 9, -8, -7, -2, -13, 5, -6)$

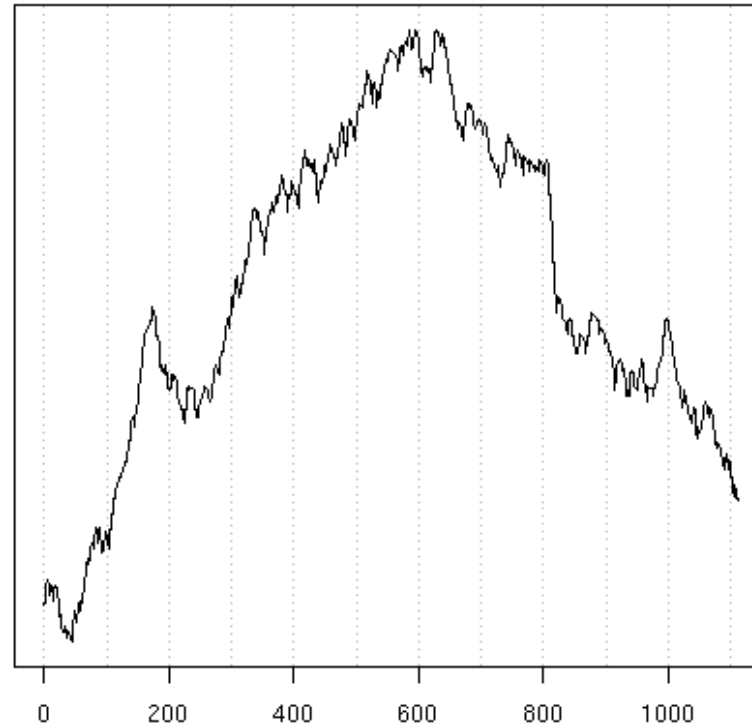
**An optimal sequence** that does not break common intervals may not exist (it is possible to relax the constraint to accept at most  $M$  interval breaks, but this also leads to an asymmetric approach).

# Biological constraints & Applications

## Terminus symmetry

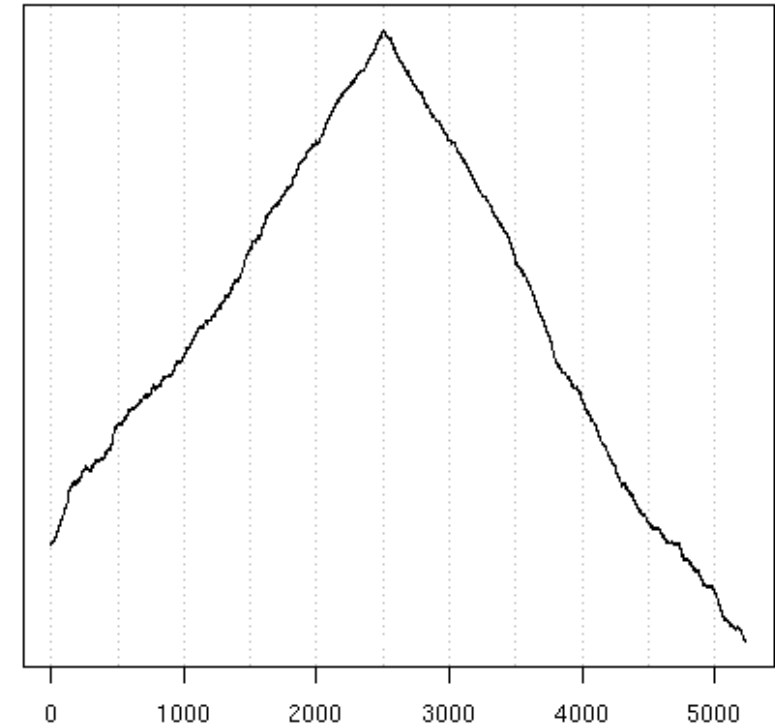


*Rickettsia prowazekii*



Map position in Kb

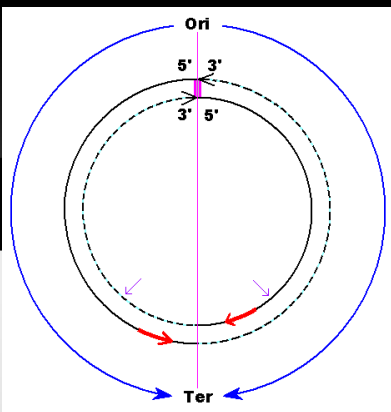
*Bacillus anthracis* Ames



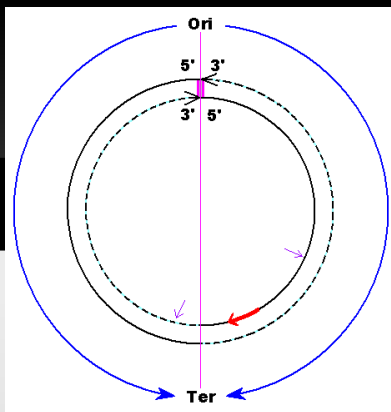
Map position in Kb

Oriloc - Prediction of origin and terminus of replication in bacteria  
(Frank & Lobry, 2000)

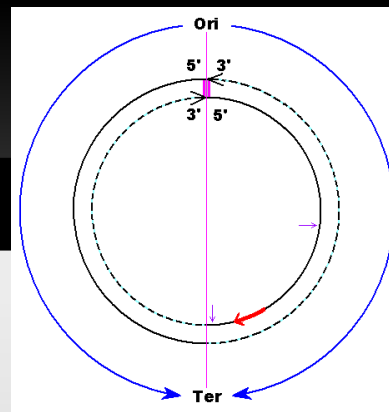
<http://pbil.univ-lyon1.fr/software/Oriloc/oriloc.html>



Symmetric

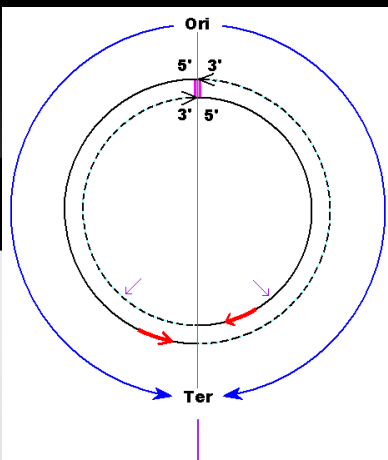


Asymmetric

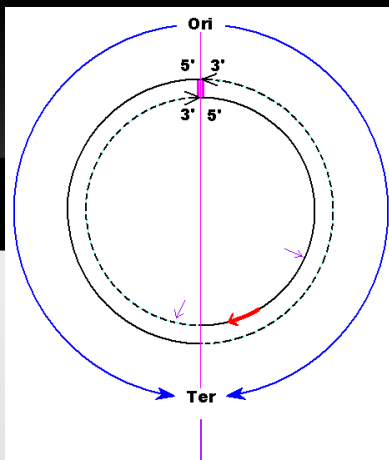


External

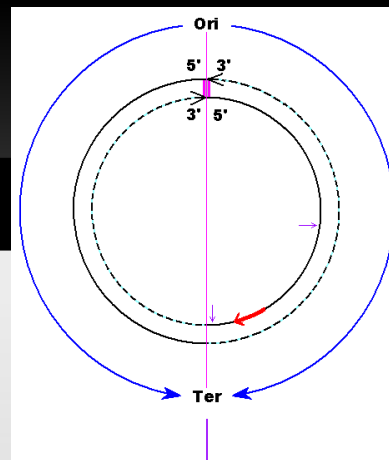
(Eisen et al,  
Genome Biology,  
2000)



Symmetric

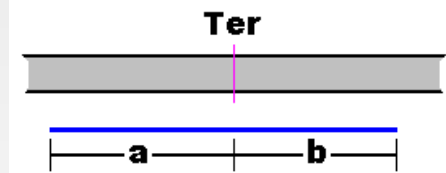
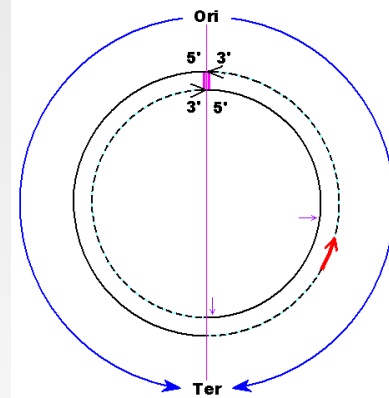
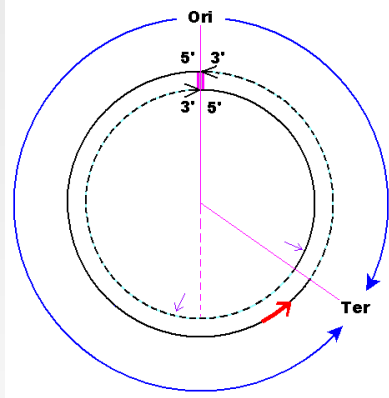
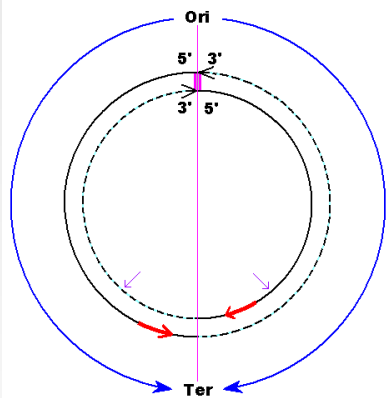
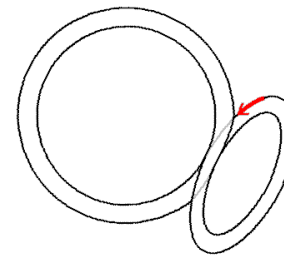
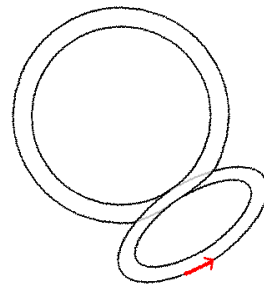
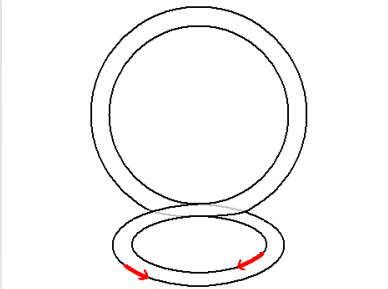


Asymmetric

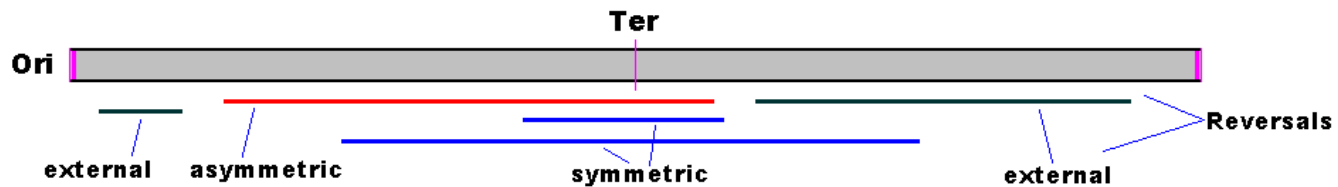


External

(Eisen et al,  
Genome Biology,  
2000)



symmetry  
rate ( $\leq 1$ ):  
 $\frac{\min(a,b)}{\max(a,b)}$



Evolution of *Rickettsia* bacterium

*Rickettsia* bacteria are  
intracellular parasites:  
conserved with reductive  
evolution

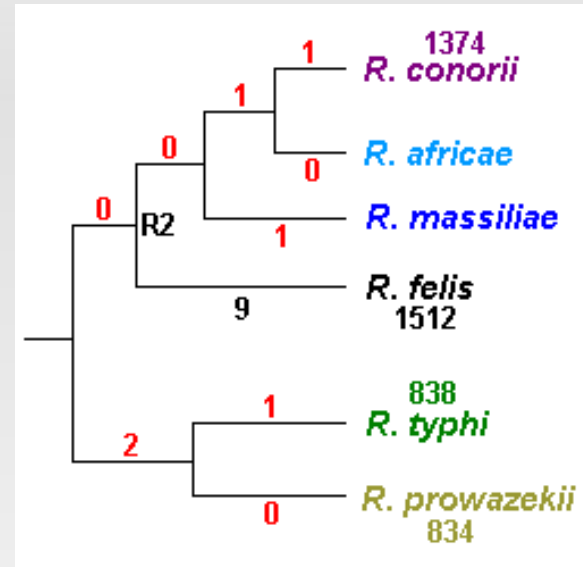
# Biological constraints & Applications

## Evolution of *Rickettsia* bacterium

*Rickettsia* bacteria are intracellular parasites: conserved with reductive evolution

Most of the known *Rickettsia* genomes are closely related

(Raoult and colleagues, 2005, 2006; Blanc et al., 2007)



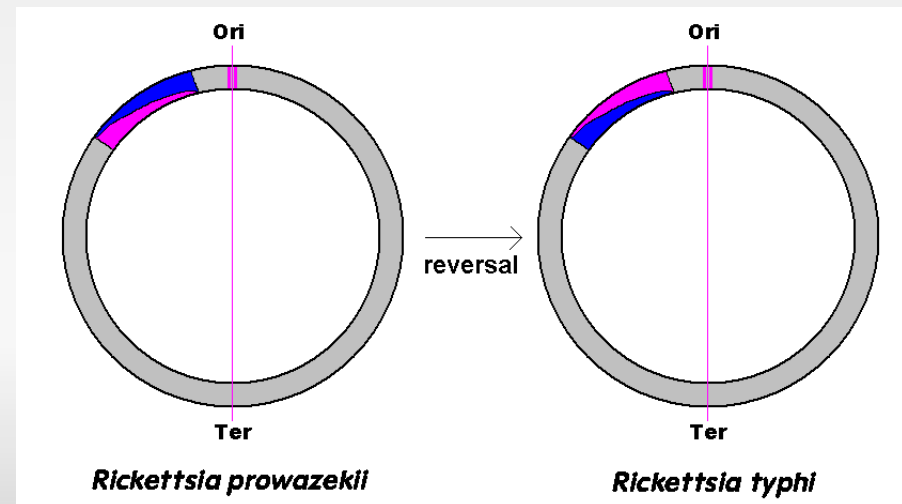
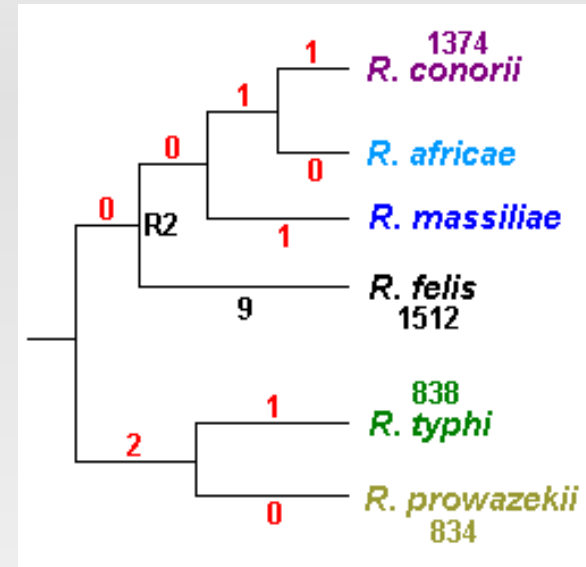
# Biological constraints & Applications

## Evolution of *Rickettsia* bacterium

*Rickettsia* bacteria are intracellular parasites: conserved with reductive evolution

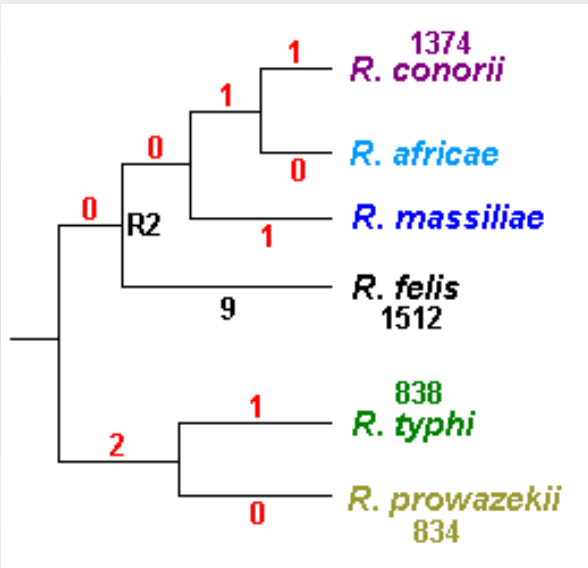
Most of the known *Rickettsia* genomes are closely related

(Raoult and colleagues, 2005, 2006; Blanc et al., 2007)



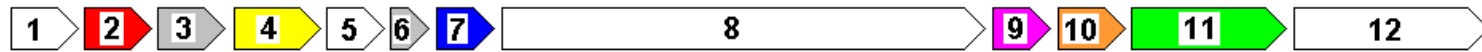
# Biological constraints & Applications

## Evolution of *Rickettsia* bacterium



Reconstructed ancestor R2 (Blanc et al, 2007):

R2

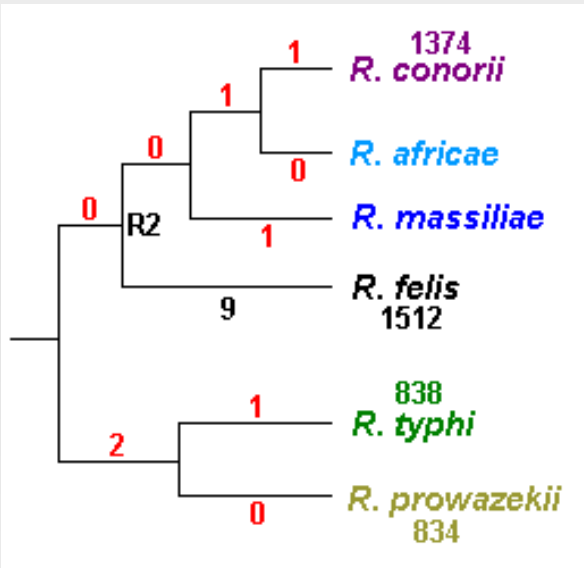


*R. felis*



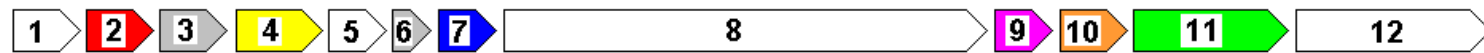
# Biological constraints & Applications

## Evolution of *Rickettsia* bacterium

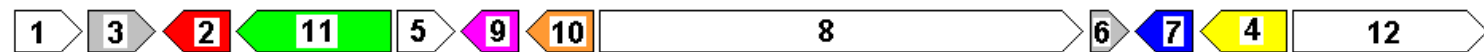


Reconstructed ancestor R2 (Blanc et al, 2007):

R2



*R. felis*

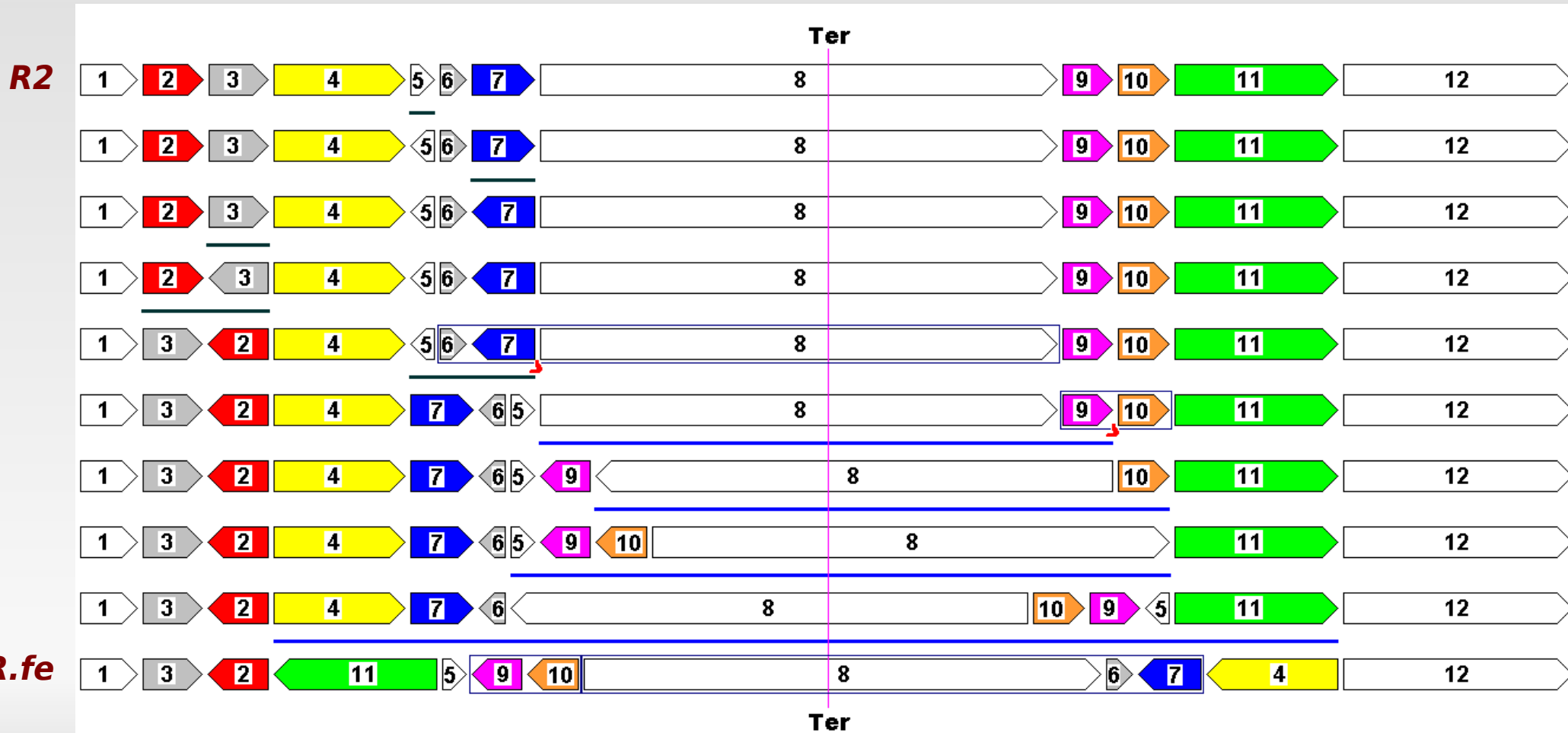


One arbitrary optimal sorting sequence was proposed by Blanc et al., (2007), obtained with GRIMM.

# Biological constraints & Applications

## Evolution of *Rickettsia* bacterium

Sequence proposed by Blanc et al. (2007) (four symmetric reversals):



**Biological constraints & Applications**

# **Evolution of *Rickettsia* bacterium**

But there are 546840 optimal sorting sequences...

# Biological constraints & Applications

## Evolution of *Rickettsia* bacterium

But there are 546840 optimal sorting sequences...

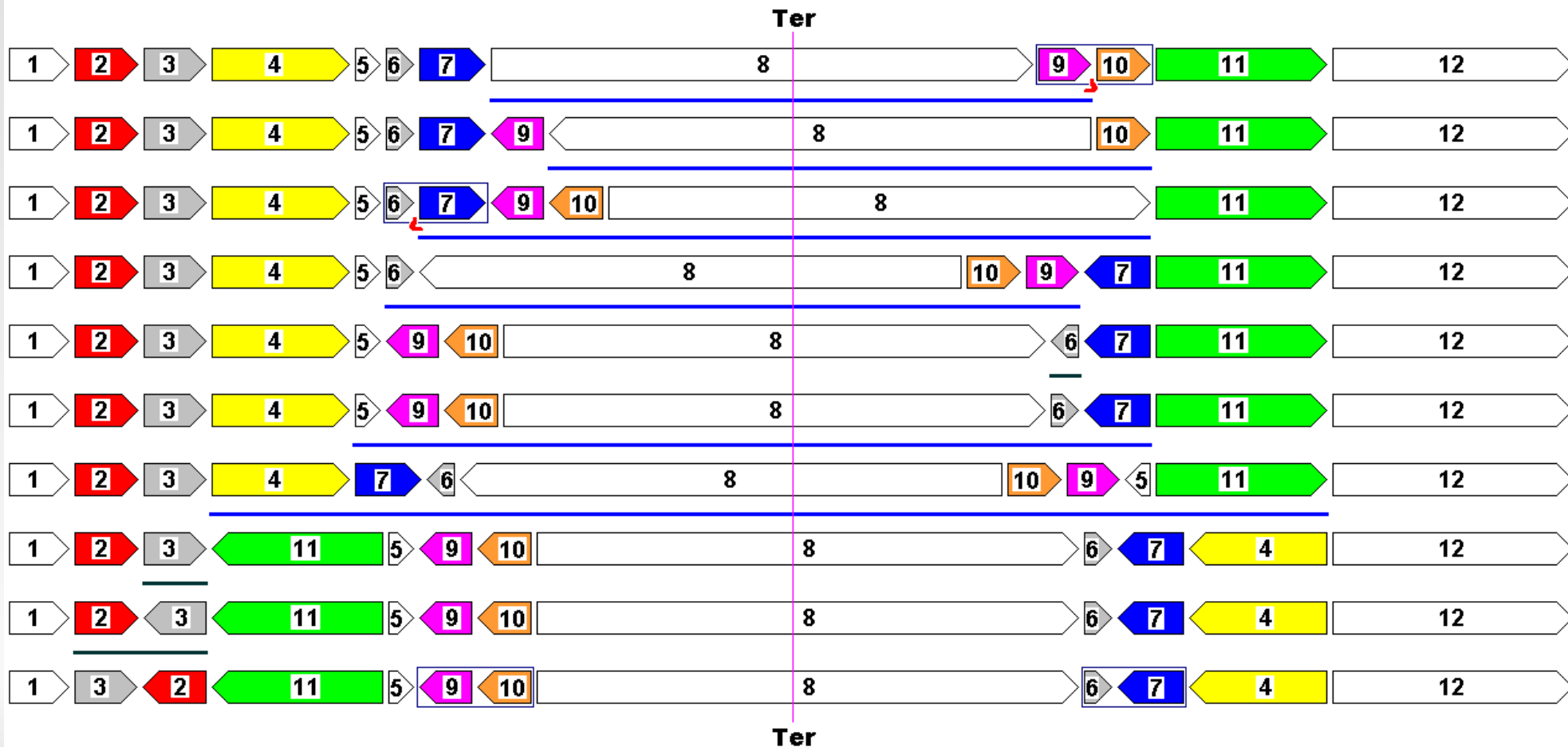
	Trace representative	# sol	elim
*	1. {2,3}{3}{4,...,11}{5}{5,8,...,10}{7}{8,10}{5,...,7}{8,9}	0	90720
	2. {2,3}{3}{4,...,11}{5,...,10}{6}{6,...,8,10}{6,8}{6,...,9}{7,8}	45360	45360
	3. {2,3}{3}{4,...,11}{5,...,10}{6}{6,8,...,10}{8,10}{7,...,10}{8,9}	45360	45360
	4. {2,3}{3}{4,...,11}{5,...,10}{6,...,8,10}{7}{9}{6,7,9}{8,9}	0	60480
	5. {2,3}{3}{4,...,11}{5,...,10}{6,8}{9}{10}{6,9,10}{7,...,10}	0	60480
	6. {2,3}{3}{4,...,11}{5,...,10}{7}{8,10}{10}{6,7,10}{6,...,9}	0	60480
	7. {2,3}{3}{4,...,11}{5,9,10}{7}{9}{10}{5,8}{5,...,7}	0	60480
	8. {2,3}{3}{4,...,11}{5,8,...,10}{5,9,10}{7}{5,...,7,9,10}{6,...,8,10}{6,...,9}	0	9072
	9. {2,3}{3}{4,...,11}{5,8,...,10}{6}{8,10}{5,6,8,9}{5,7,...,9}{6,...,9}	0	6048
	10. {2,3}{3}{4,...,11}{5,9,10}{6,8}{10}{5,6,9}{5,7,...,9}{6,...,9}	0	6048
	11. {2,3}{3}{4,...,11}{6}{6,8,...,10}{5,6,8,10}{5,6,8,9}{5,...,8}{7,8}	6048	0
	12. {2,3}{3}{4,...,11}{5}{6,8,...,10}{5,6,8,10}{5,7,9}{6,7,9}{8,9}	0	3024
	13. {2,3}{3}{4,...,11}{6,8}{6,9,10}{7,8}{5,6,10}{5,6,9}{5,...,8}	0	2520
	<b>Total</b>	<b>96768</b>	<b>450072</b>

# Biological constraints & Applications

## Evolution of *Rickettsia* bacterium

An alternative scenario with six symmetric reversals:

*R2*

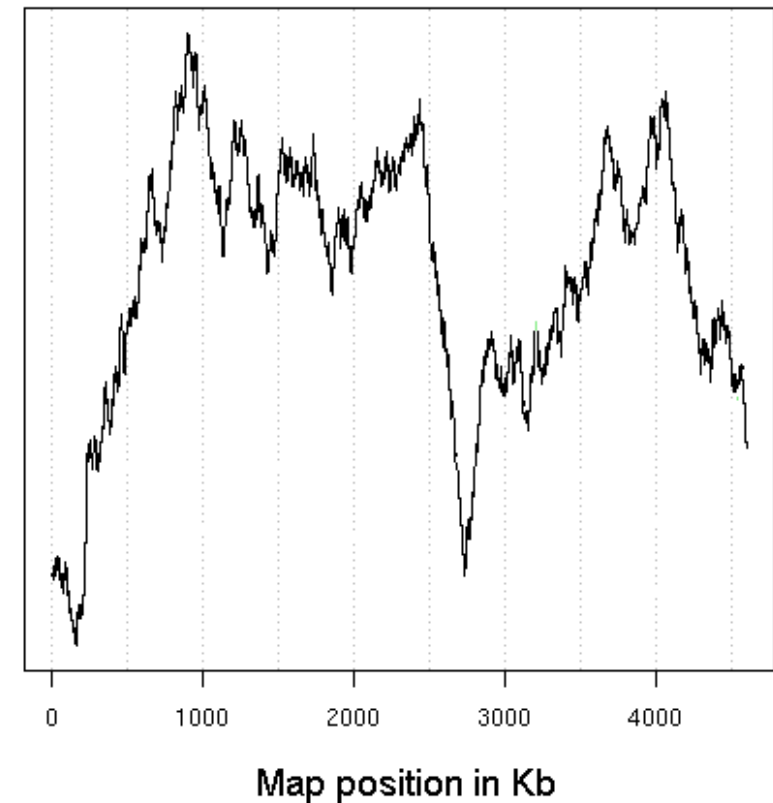


# Biological constraints & Applications

## Evolution of *Rickettsia* bacterium

- One should consider whether this approach is adequate to the analyzed organisms.

**Yersinia pestis biovar Mediaevalis**



The algorithm for enumerating all the traces is  
integrated to the **baobabLuna framework**

**On-line download:** <http://pbil.univ-lyon1.fr/software/luna/>

The algorithm for enumerating all the traces is integrated to the **baobabLuna framework**

**On-line download:** <http://pbil.univ-lyon1.fr/software/luna/>

## **Optimization of memory use:**

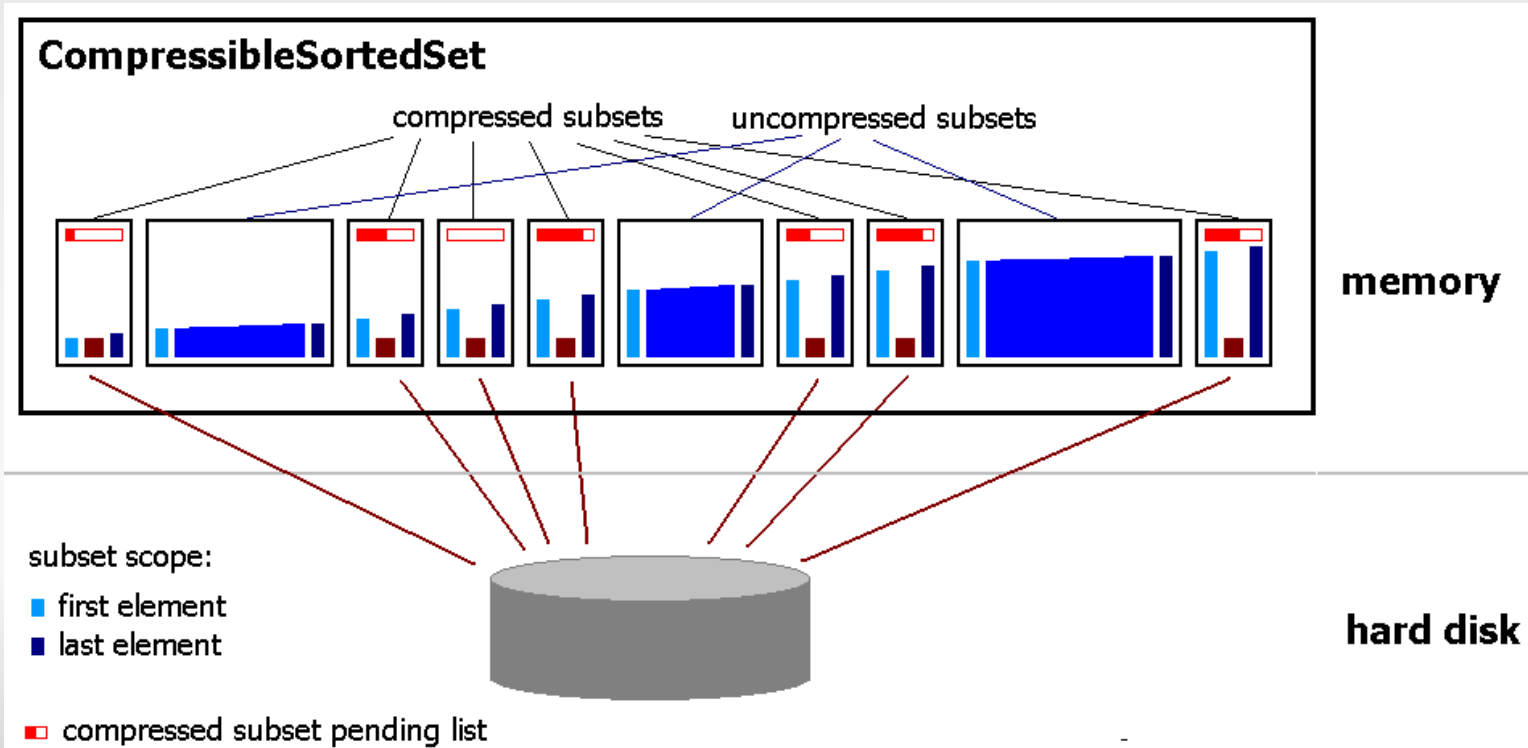
When analyzing traces, for a reversal distance of  $\sim 15$ , we may need to handle more than 200 000 000 partial traces at one step)

The algorithm for enumerating all the traces is integrated to the **baobabLuna framework**

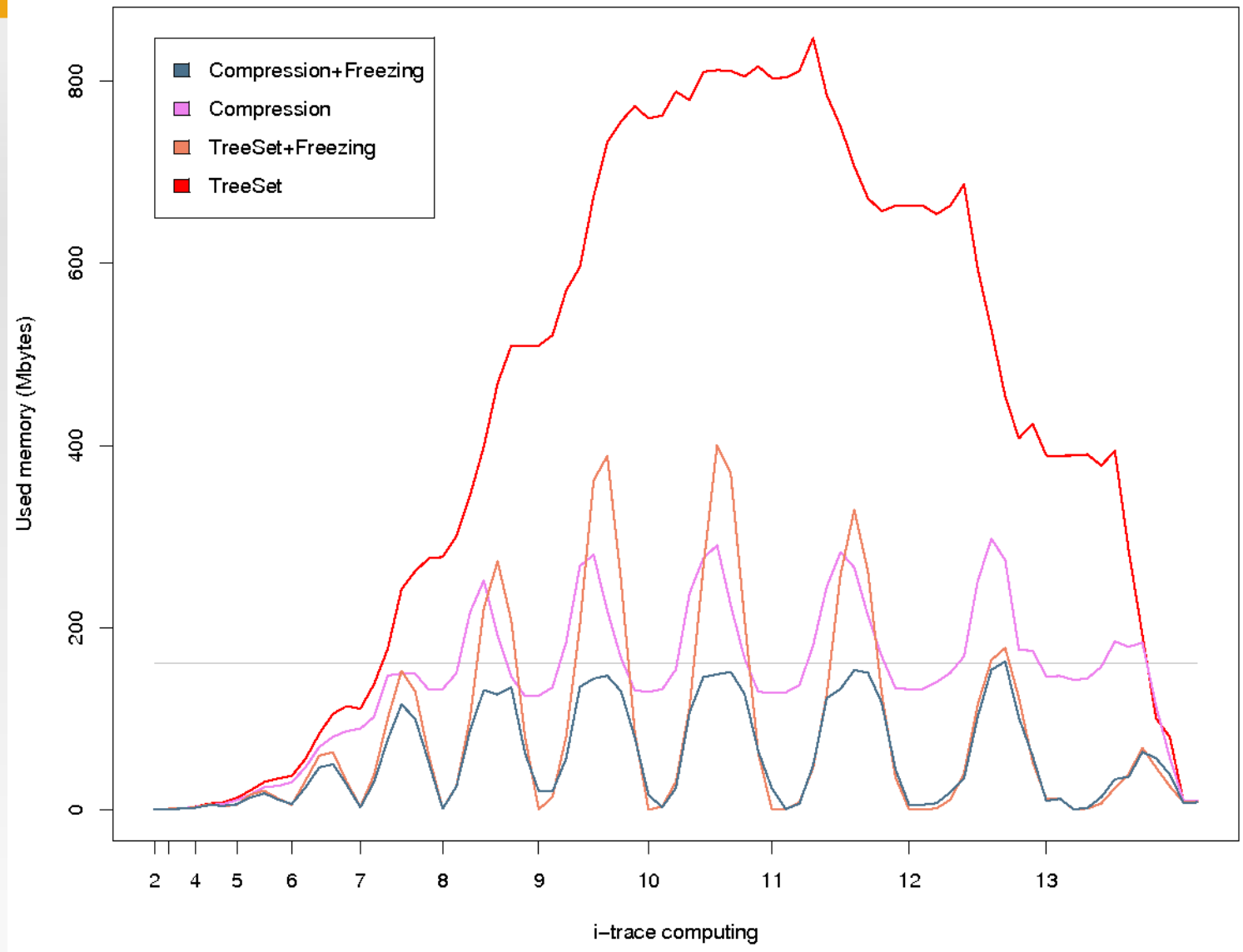
**On-line download:** <http://pbil.univ-lyon1.fr/software/luna>

## Optimization of memory use:

When analyzing traces, for a reversal distance of  $\sim 15$ , we may need to handle more than 200 000 000 partial traces at one step



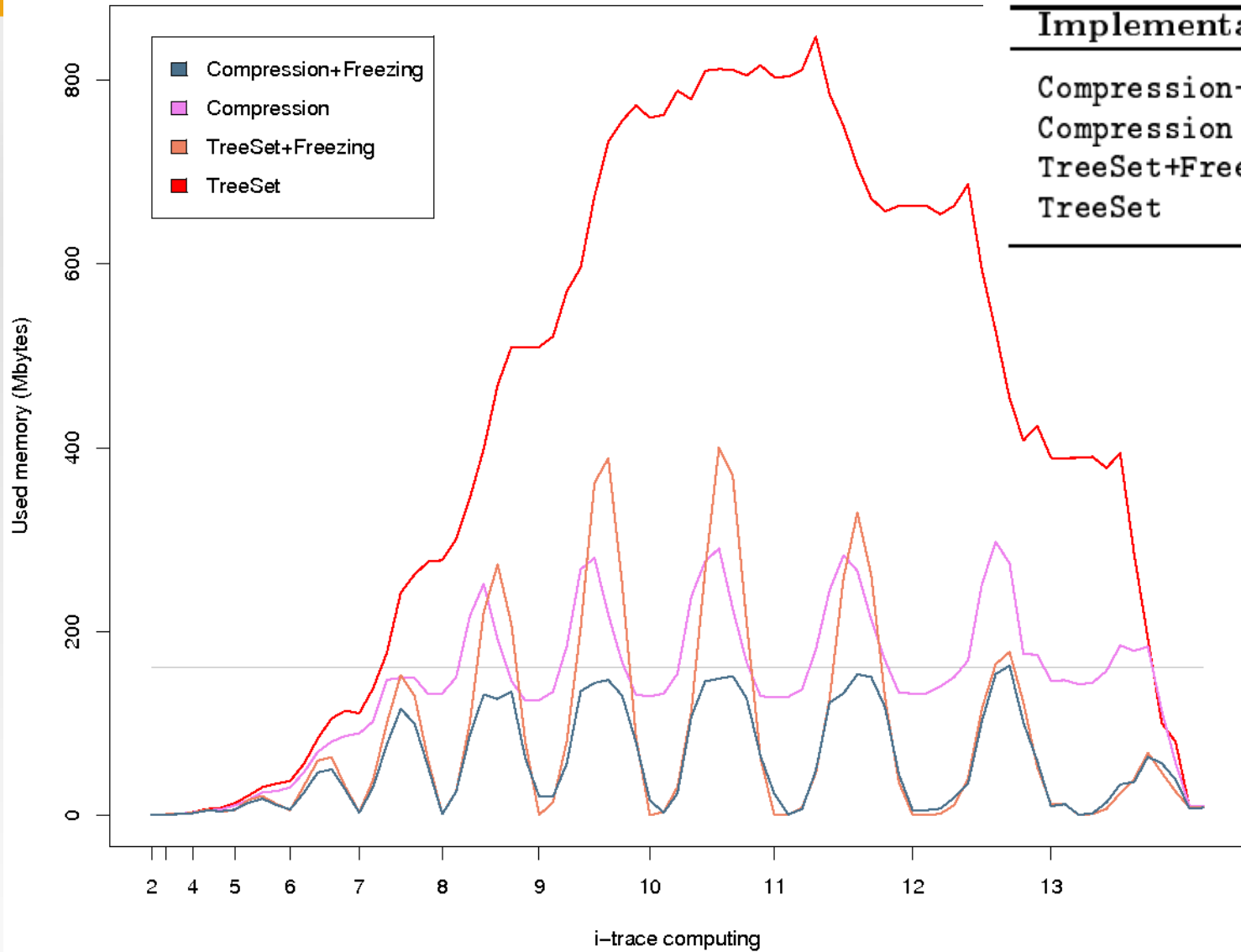
Performance of standard and test implementations with respect to memory use



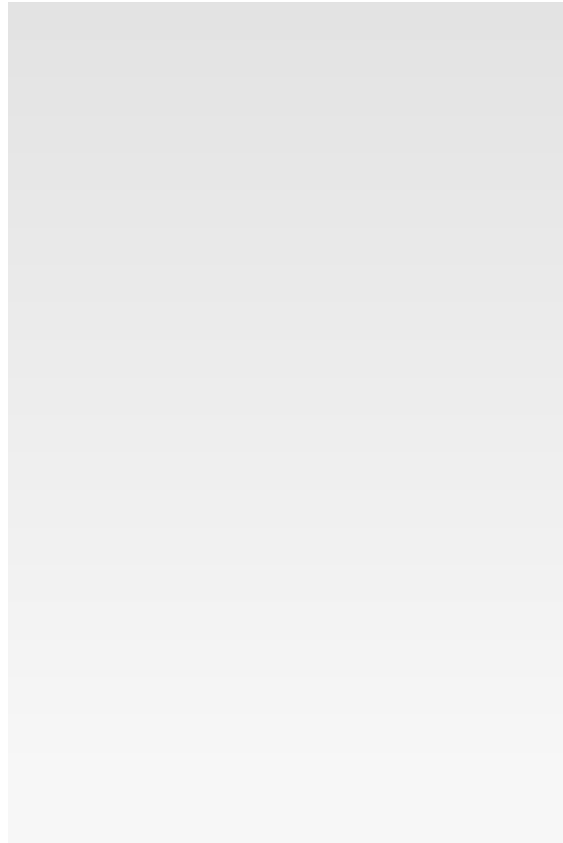
Total number of *i*-traces

<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>
690	6331	38986	172257	567851	1413316	2669032	3824570	4047048	3298406	1958839	567524

Performance of standard and test implementations with respect to memory use



Implementation	Execution time
Compression+Freezing	≈ 4.1 hours
Compression	≈ 4.1 hours
TreeSet+Freezing	≈ 4.8 hours
TreeSet	≈ 4.2 hours



Total number of *i*-traces

<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>
690	6331	38986	172257	567851	1413316	2669032	3824570	4047048	3298406	1958839	567524

# Final conclusions

# Final conclusions

An algorithm to represent the **space of all solutions** for the sorting by reversals problem

# Final conclusions

An algorithm to represent the **space of all solutions** for the sorting by reversals problem

**Use of biological constraints** to reduce the solution space (such as the common intervals)

# Final conclusions

An algorithm to represent the **space of all solutions** for the sorting by reversals problem

**Use of biological constraints** to reduce the solution space (such as the common intervals)

**Application:** study of Rickettsia bacteria evolution

# Final conclusions

An algorithm to represent the **space of all solutions** for the sorting by reversals problem

**Use of biological constraints** to reduce the solution space (such as the common intervals)

**Application:** study of Rickettsia bacteria evolution

**baobabLuna:** a java framework to deal with permutations, with optimization of memory use

**Thank you !**