LaCIM Math Dept. LaBRI

Counting and representing DCJ sorting scenarios

Marília Braga
Jens Stoye

Technische Fakultät
Universität Bielefeld

Aïda Ouangraoua
Anne Bergeron

LaCIM, UQAM
Math Department, SFU

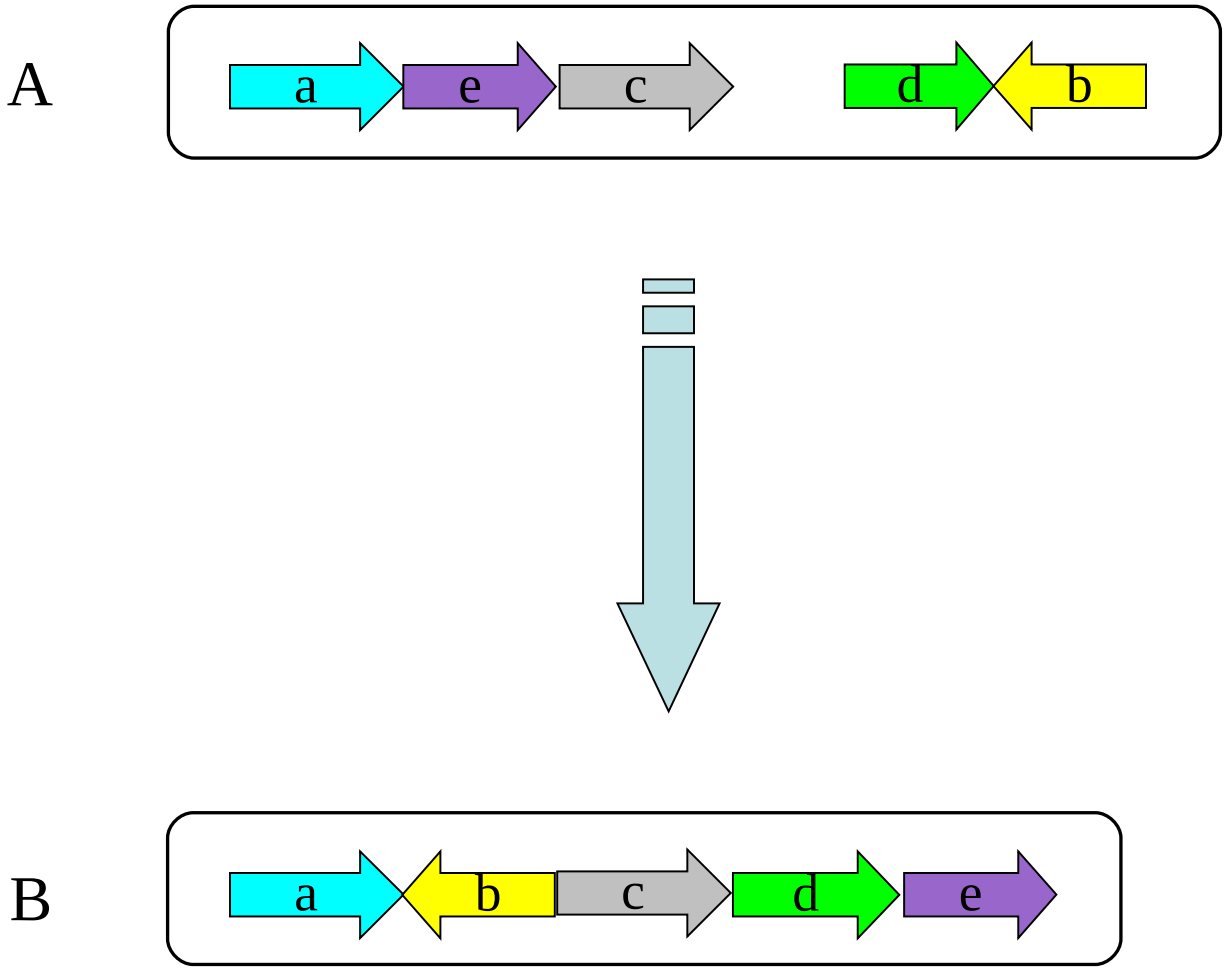
Overview

- Introduction: the DCJ model
- Counting DCJ sorting scenarios
- Representing DCJ sorting scenarios
- Conclusions and perspectives

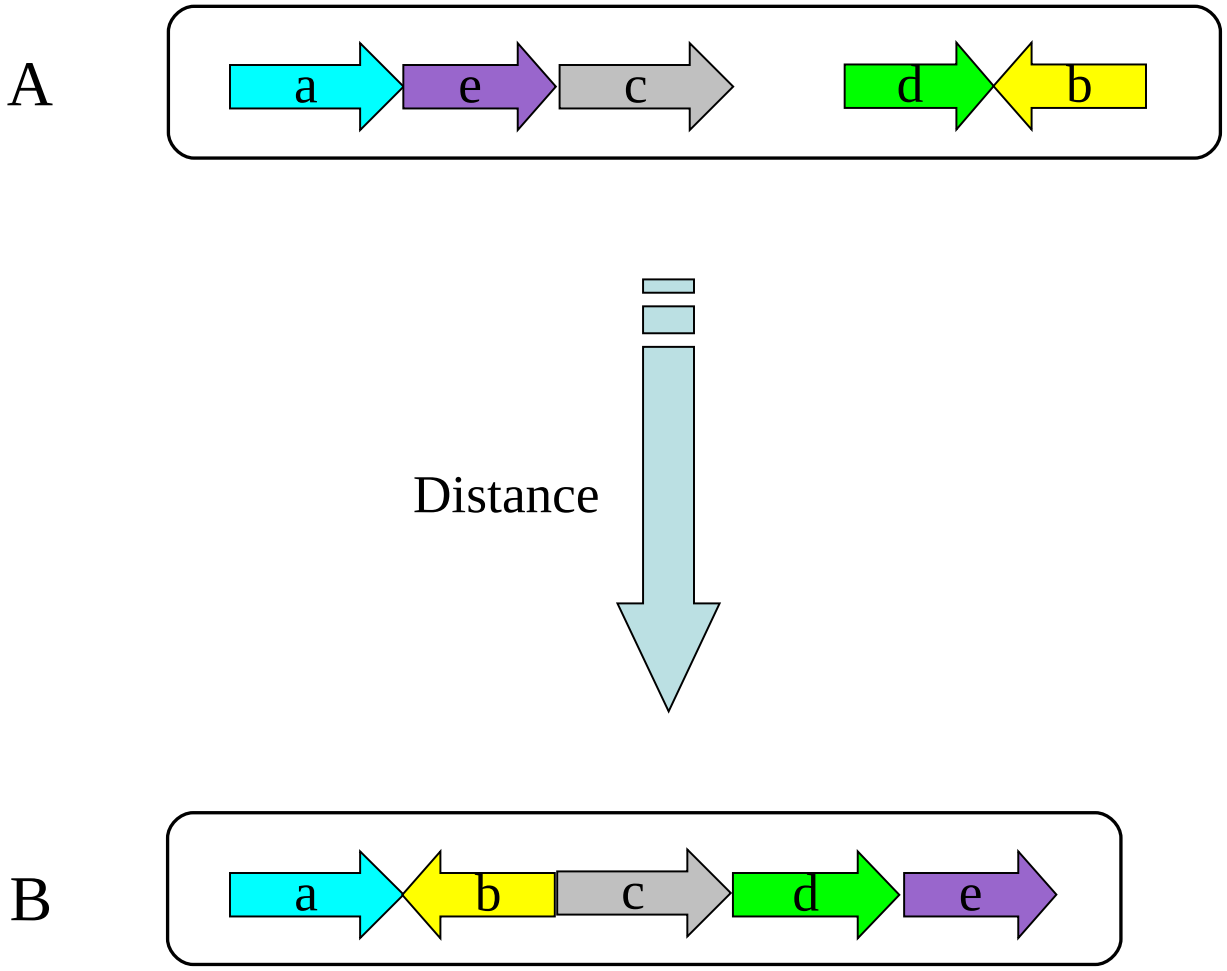
Overview

- Introduction: the DCJ model
- Counting DCJ sorting scenarios
- Representing DCJ sorting scenarios
- Conclusions and perspectives

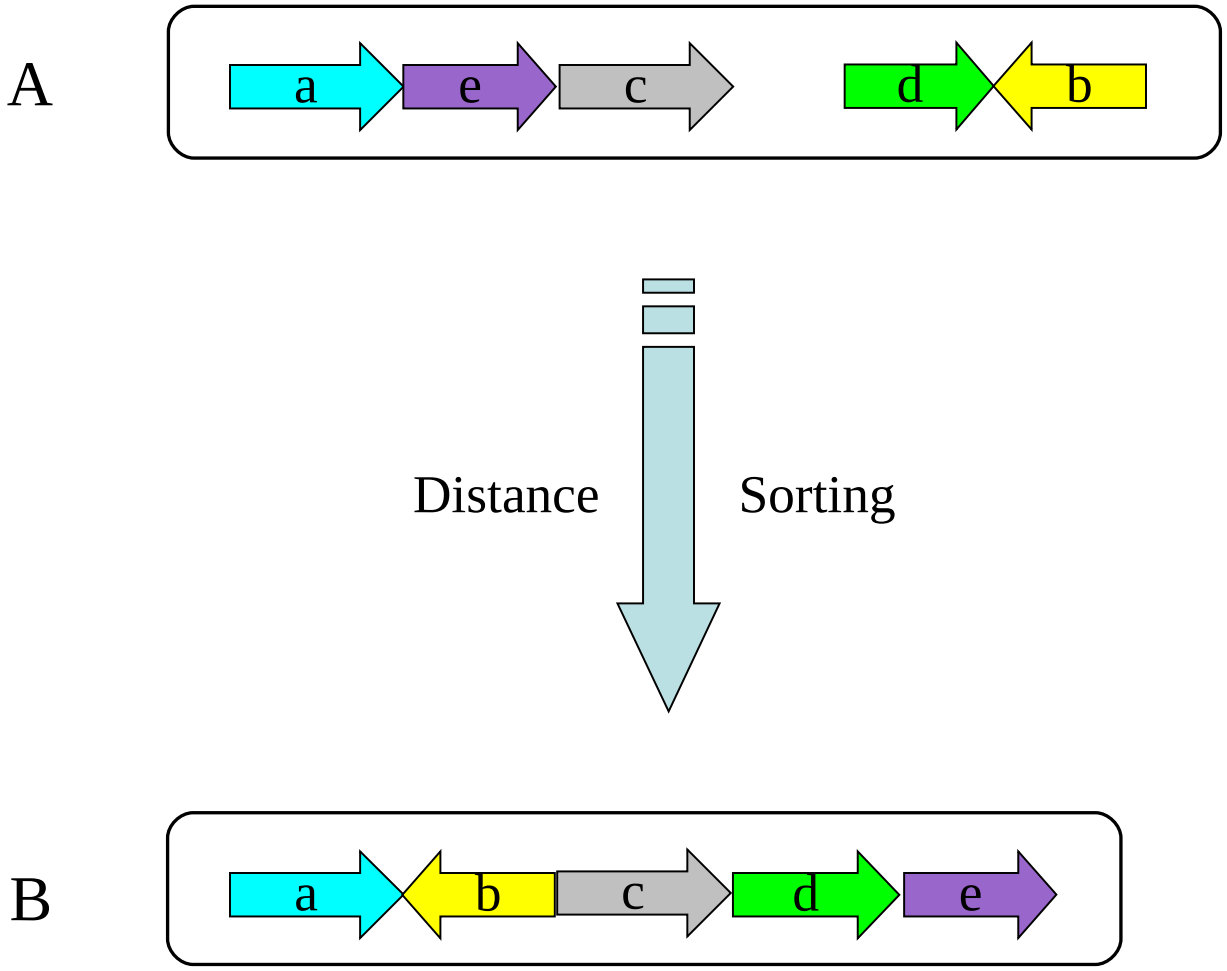
Genome rearrangements



Genome rearrangements

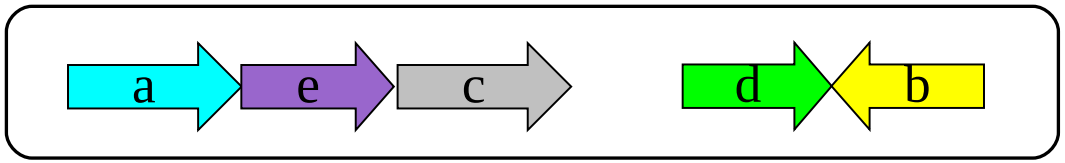


Genome rearrangements

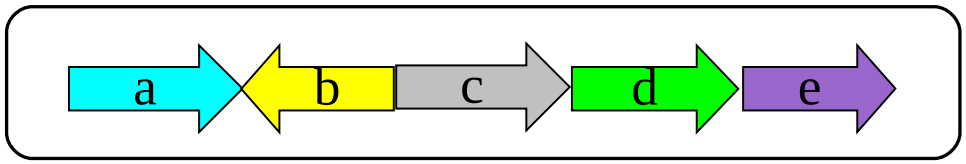


The double-cut-and-join (DCJ) model

A

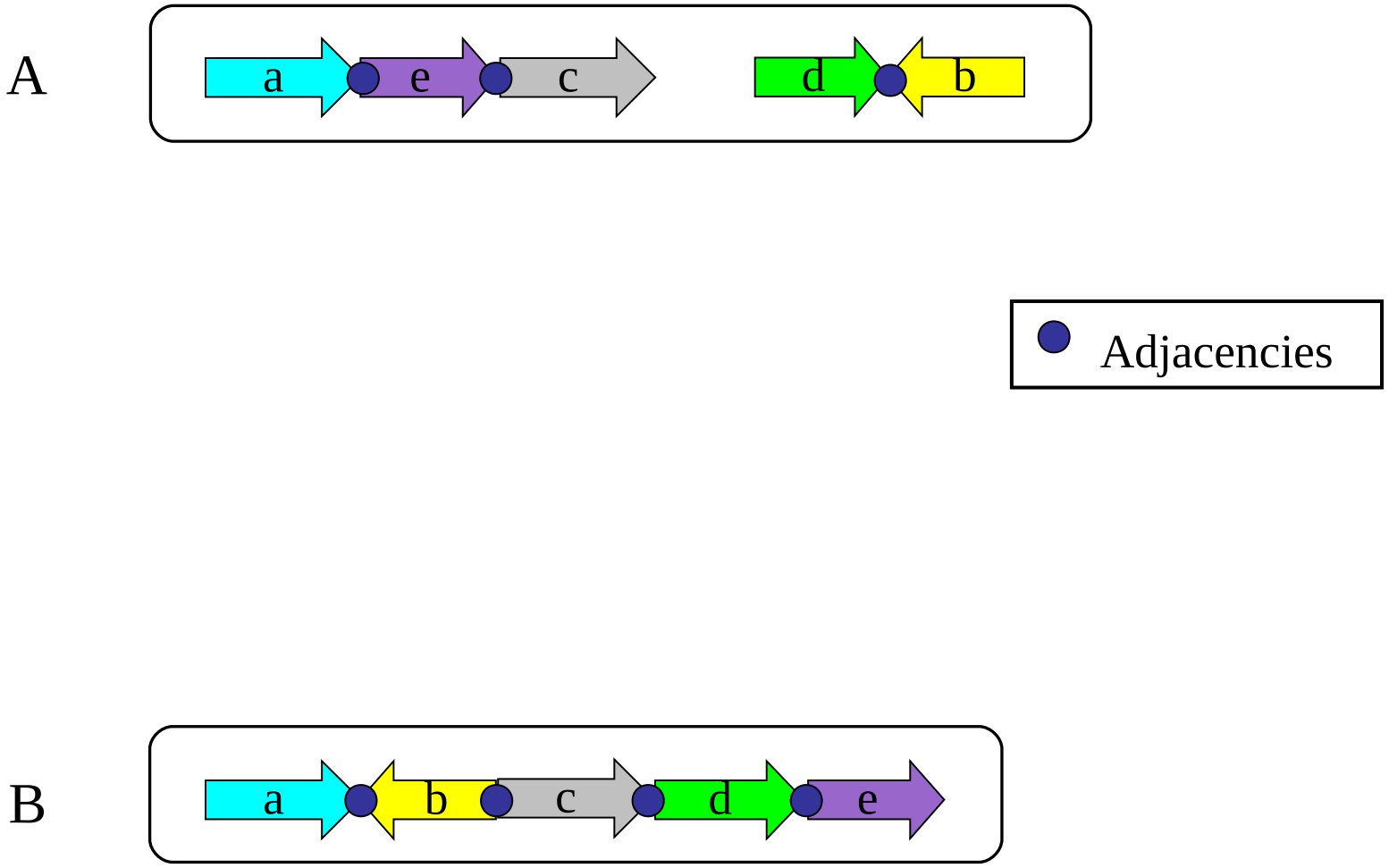


B



(Yancopoulos et al. 2005)

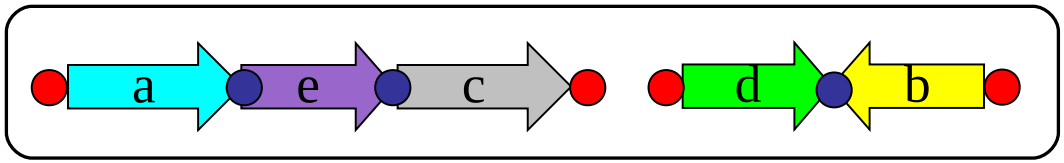
The double-cut-and-join (DCJ) model

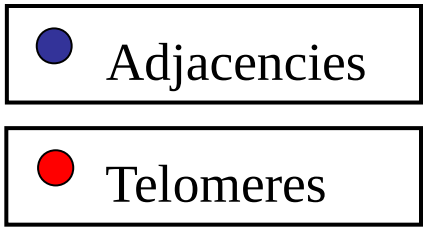


(Yancopoulos et al. 2005)

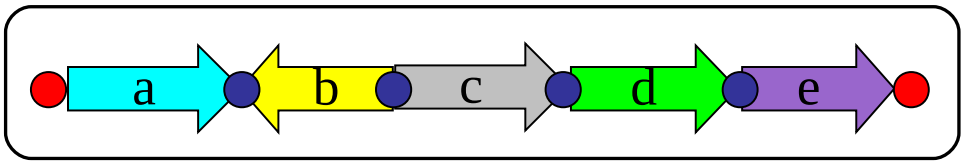
The double-cut-and-join (DCJ) model

A





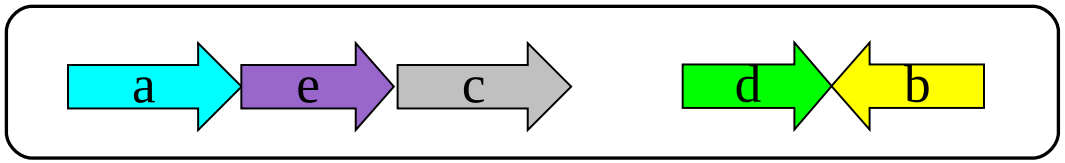
B



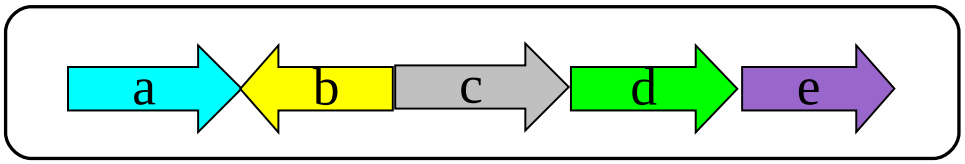
(Yancopoulos et al. 2005)

The double-cut-and-join (DCJ) model

A

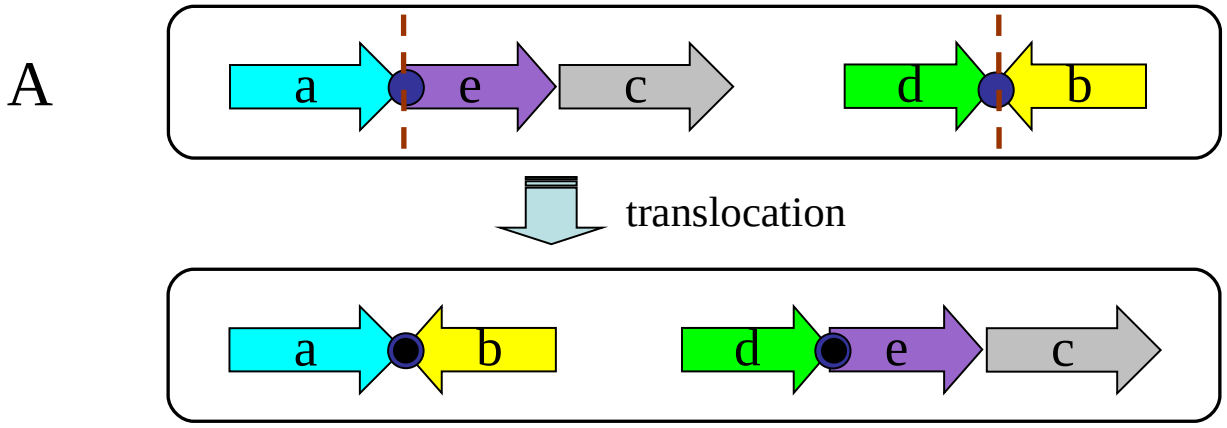


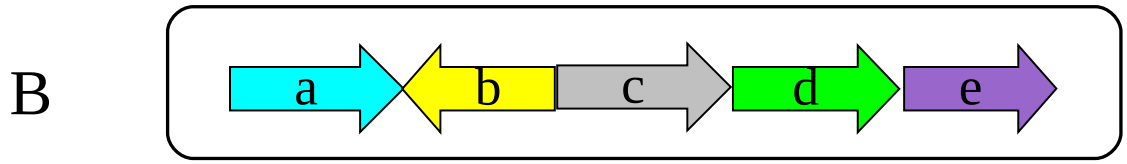
B



(Yancopoulos et al. 2005)

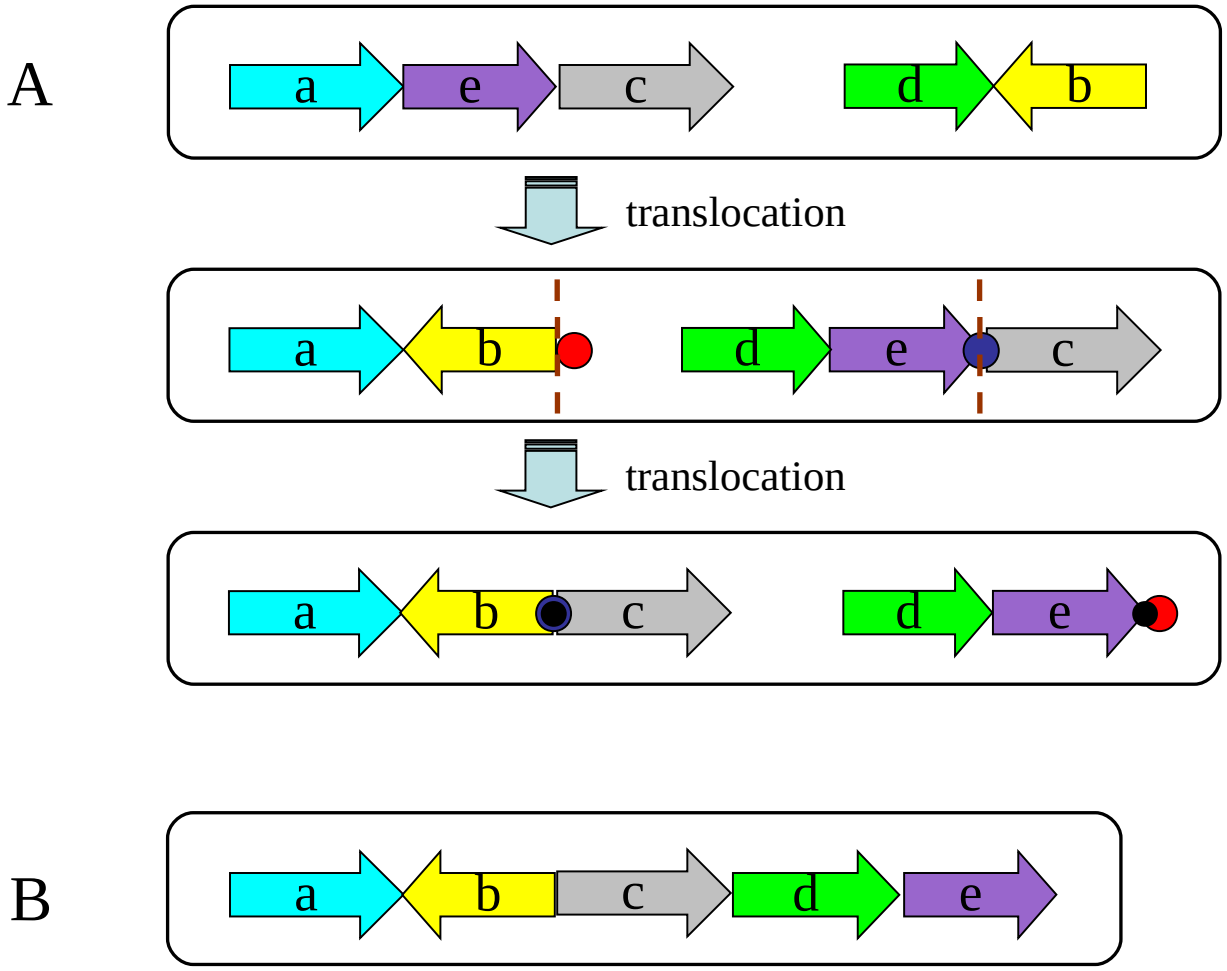
The double-cut-and-join (DCJ) model





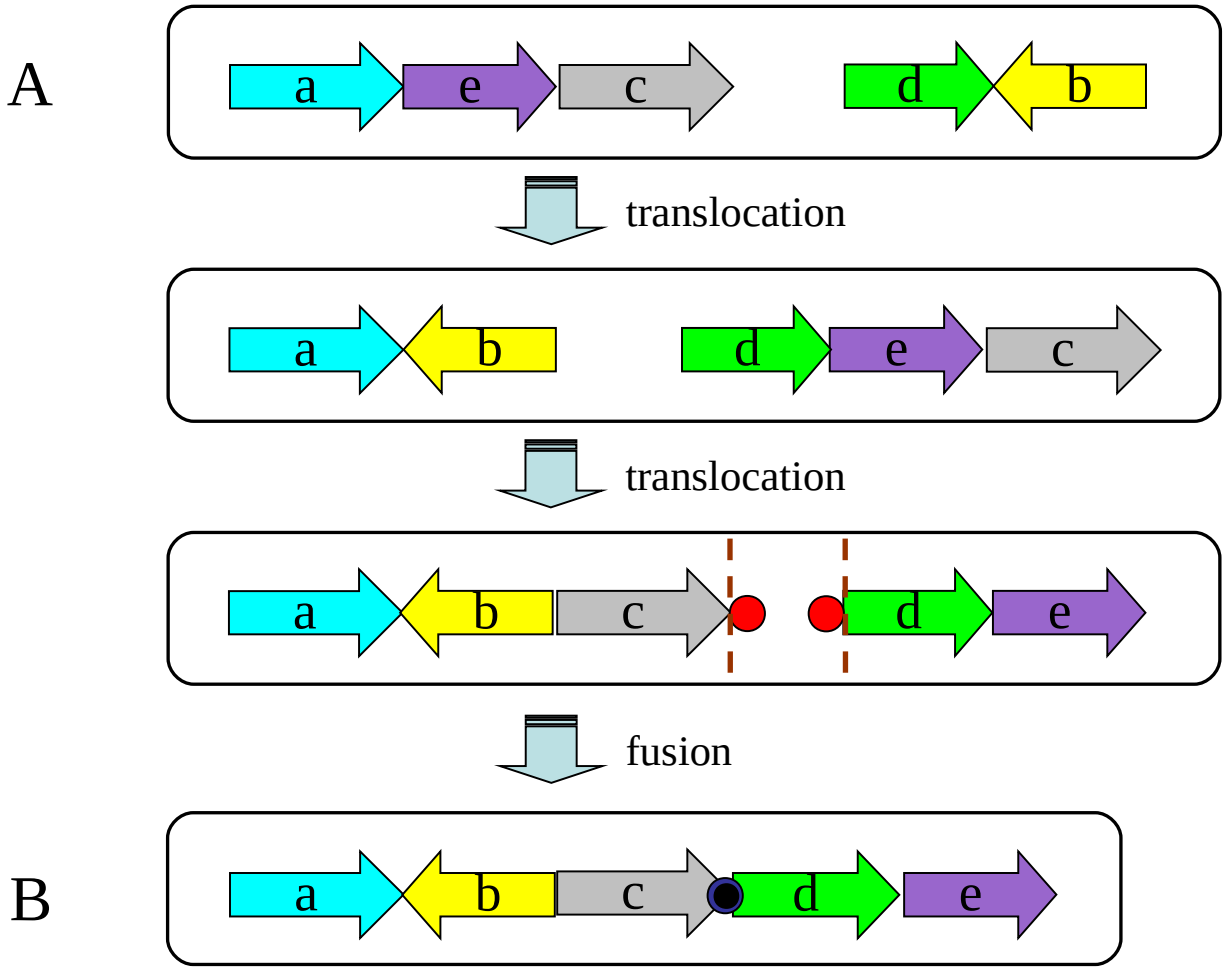
(Yancopoulos et al. 2005)

The double-cut-and-join (DCJ) model



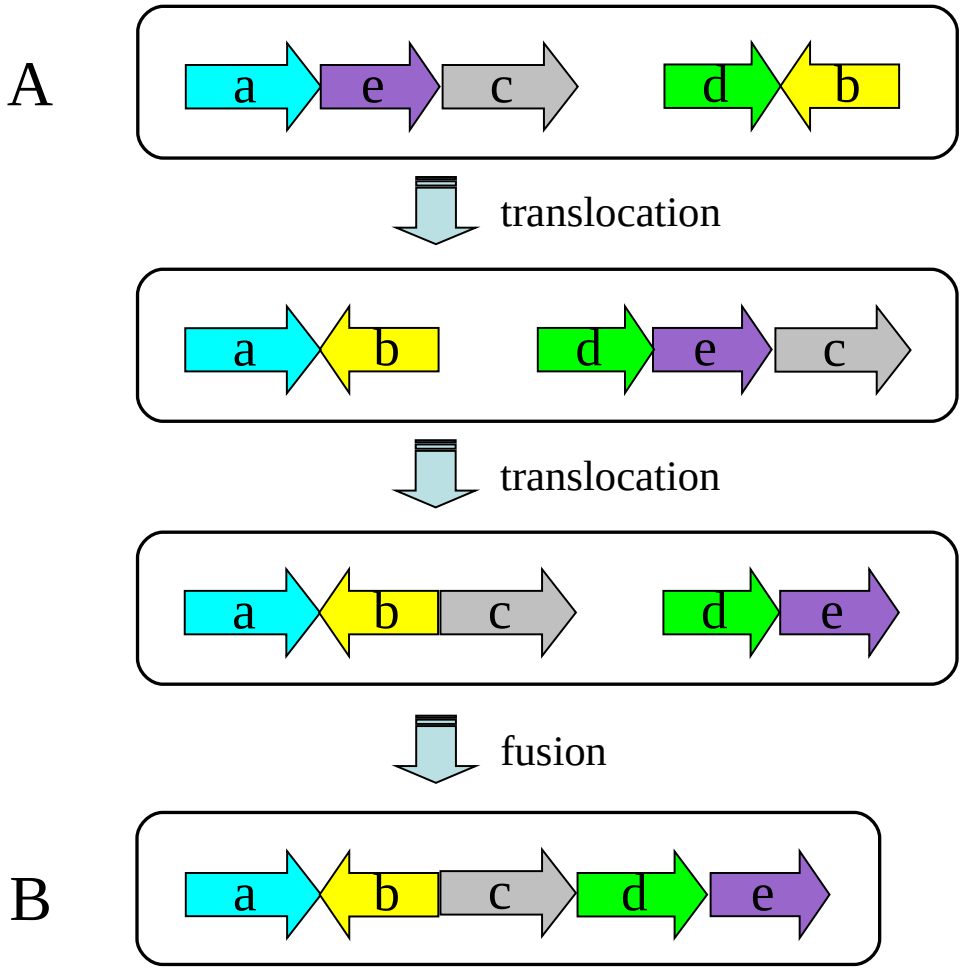
(Yancopoulos et al. 2005)

The double-cut-and-join (DCJ) model

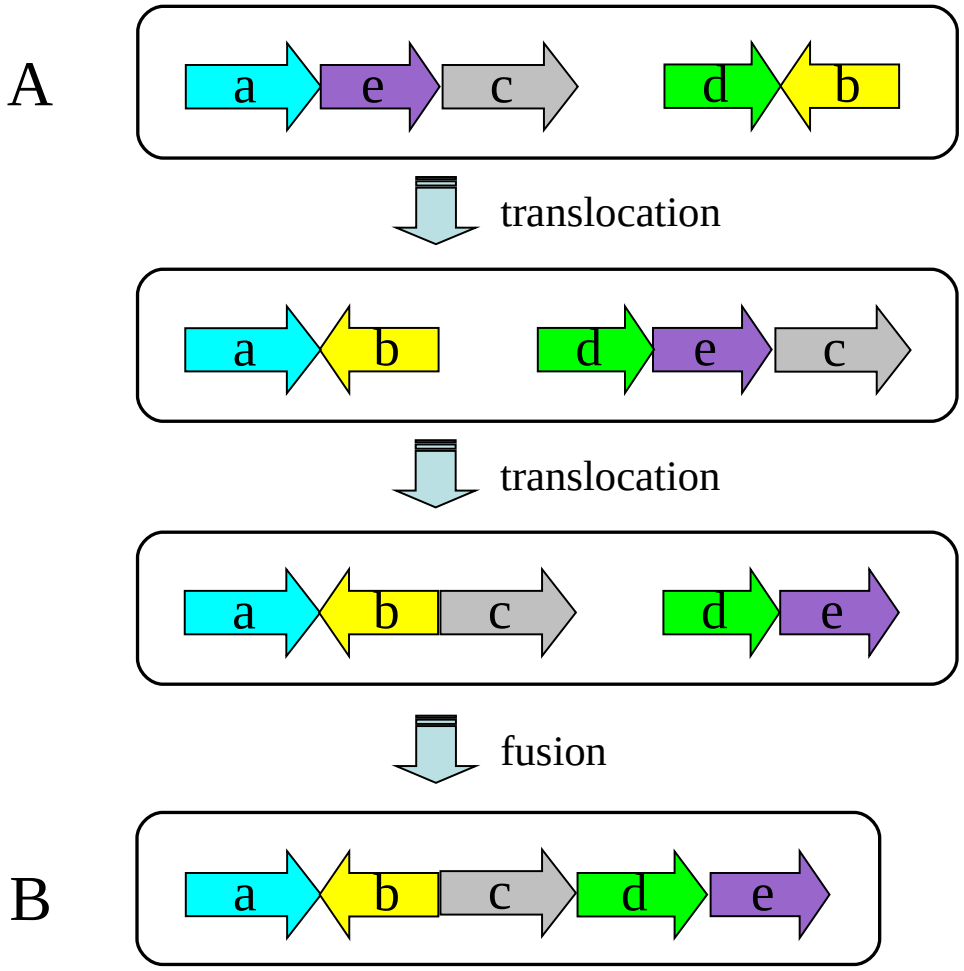


(Yancopoulos et al. 2005)

The double-cut-and-join (DCJ) model



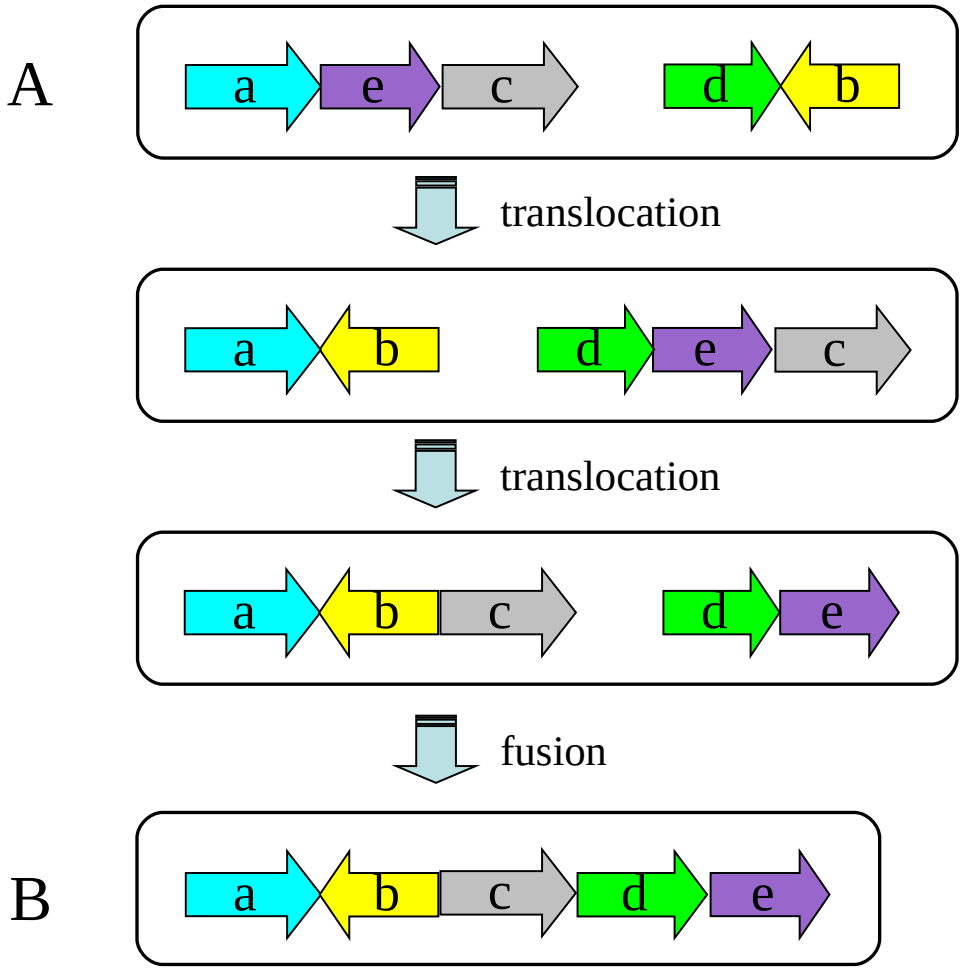
The double-cut-and-join (DCJ) model



Computing the DCJ distance and generating one sorting scenario:

Bergeron et al. 2006

The double-cut-and-join (DCJ) model



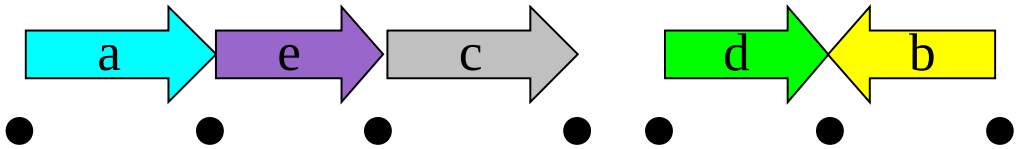
Computing the DCJ distance and generating one sorting scenario:

Bergeron et al. 2006

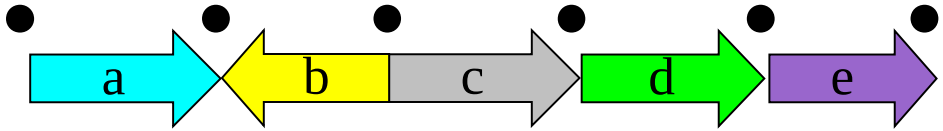
Counting and representing the space of all sorting scenarios

Adjacency Graph

A

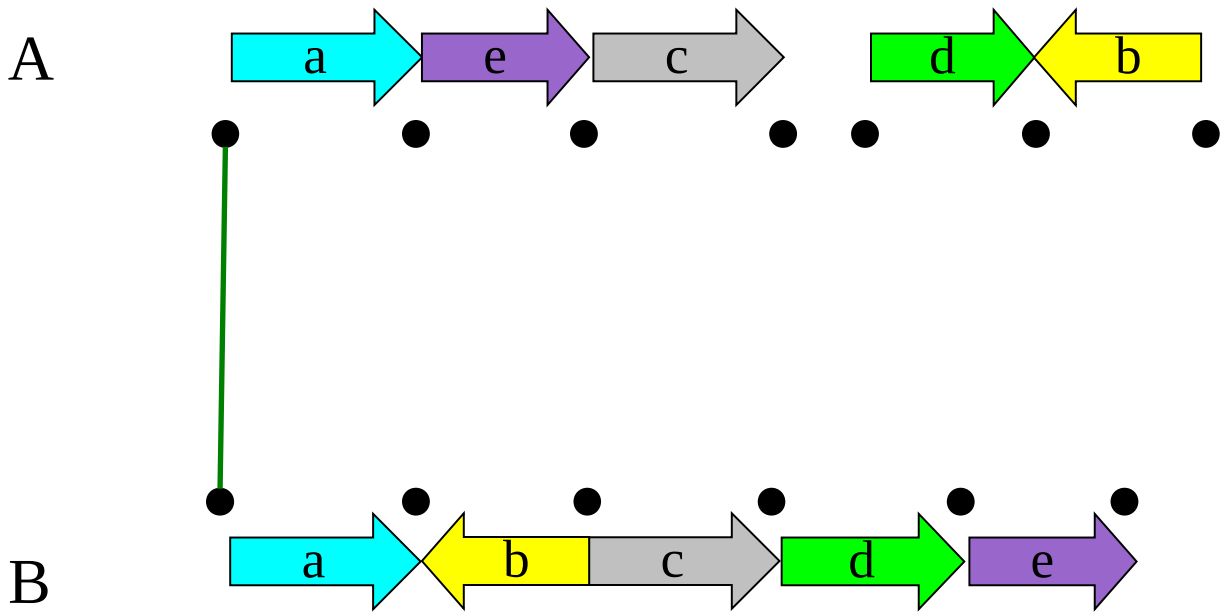


B



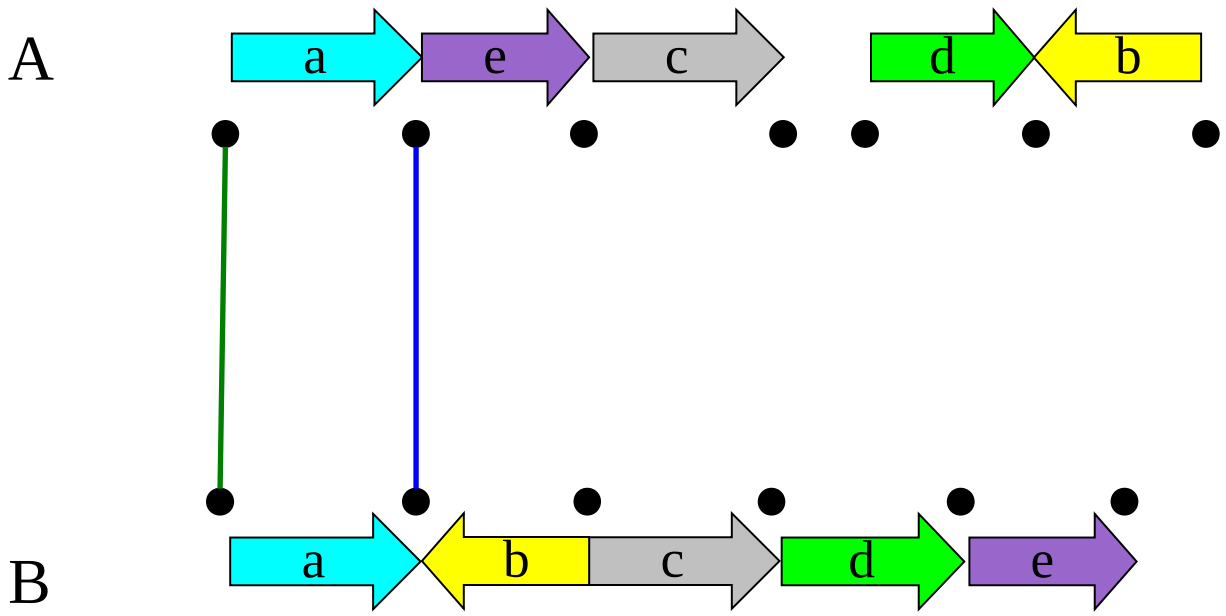
(Bergeron et al. 2006)

Adjacency Graph



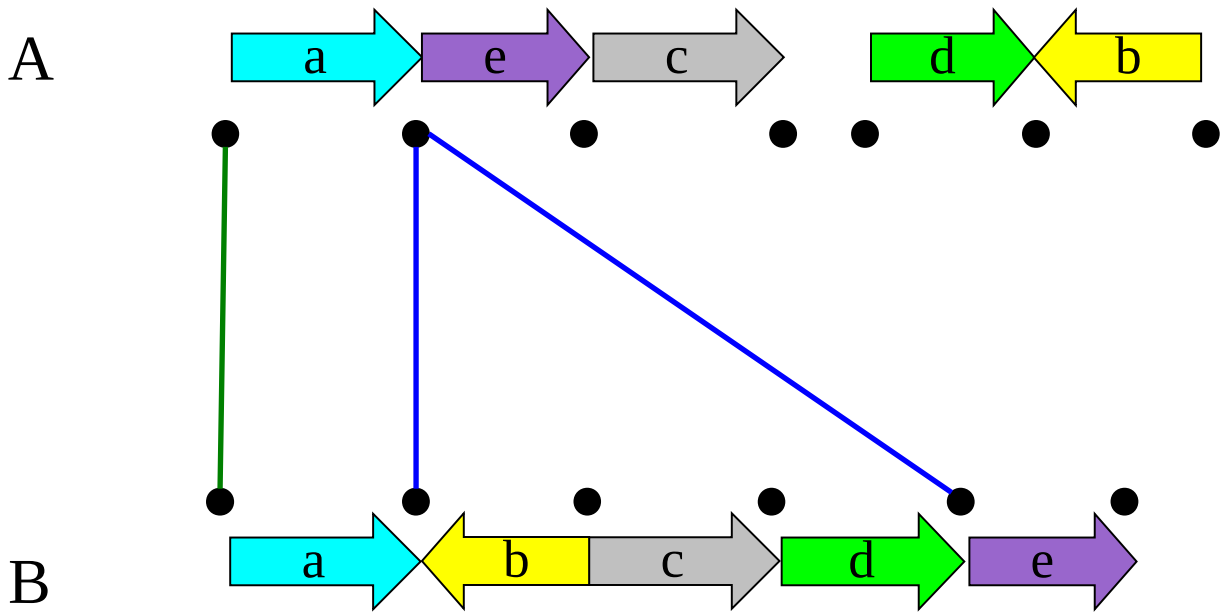
(Bergeron et al. 2006)

Adjacency Graph



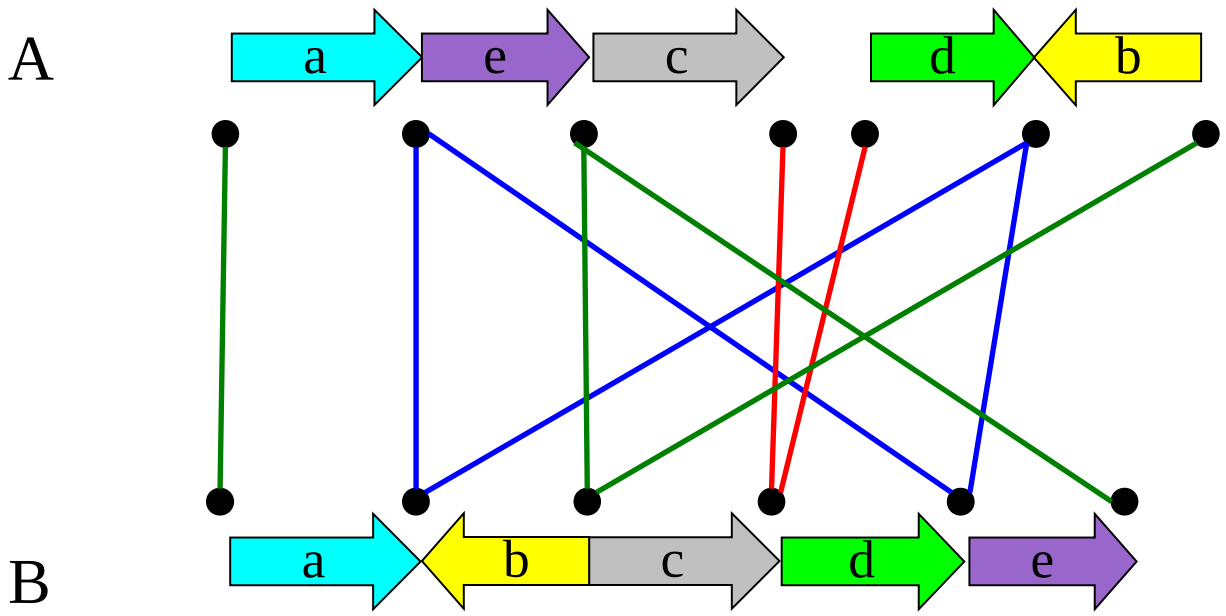
(Bergeron et al. 2006)

Adjacency Graph



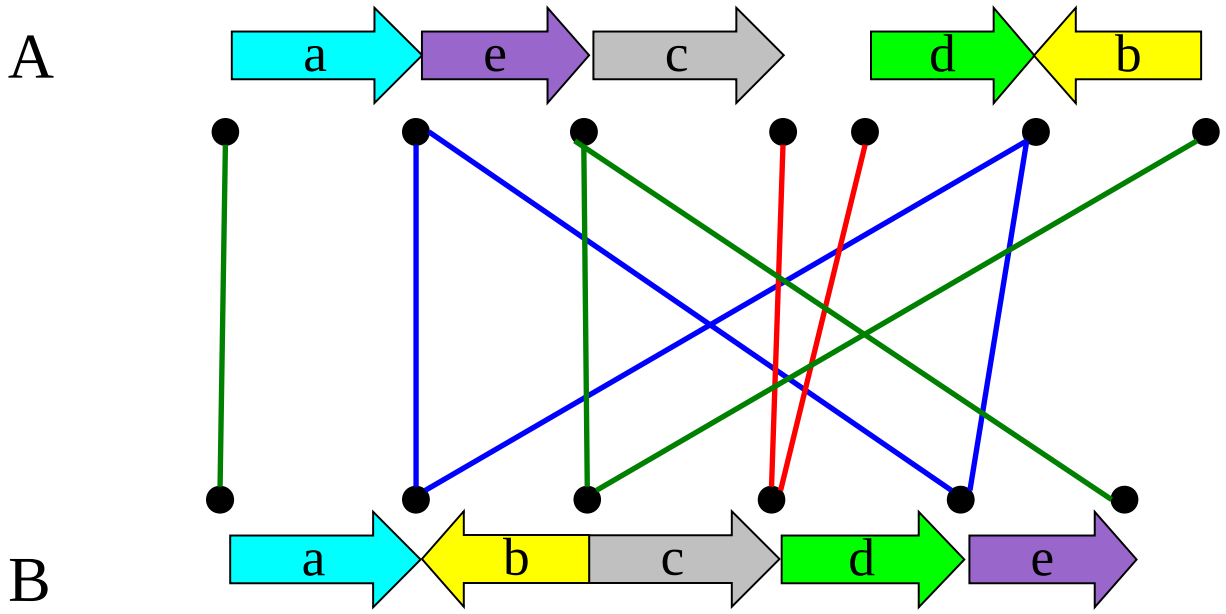
(Bergeron et al. 2006)

Adjacency Graph



(Bergeron et al. 2006)

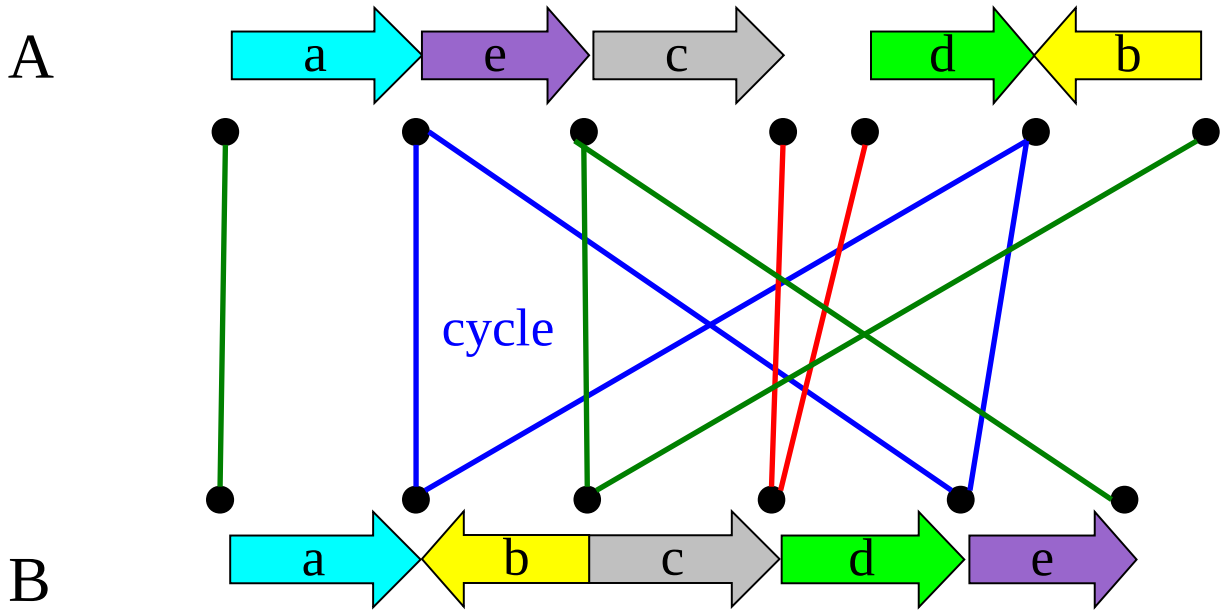
Adjacency Graph



- Bipartite graph, maximum degree equal to two: a collection of cycles and paths

(Bergeron et al. 2006)

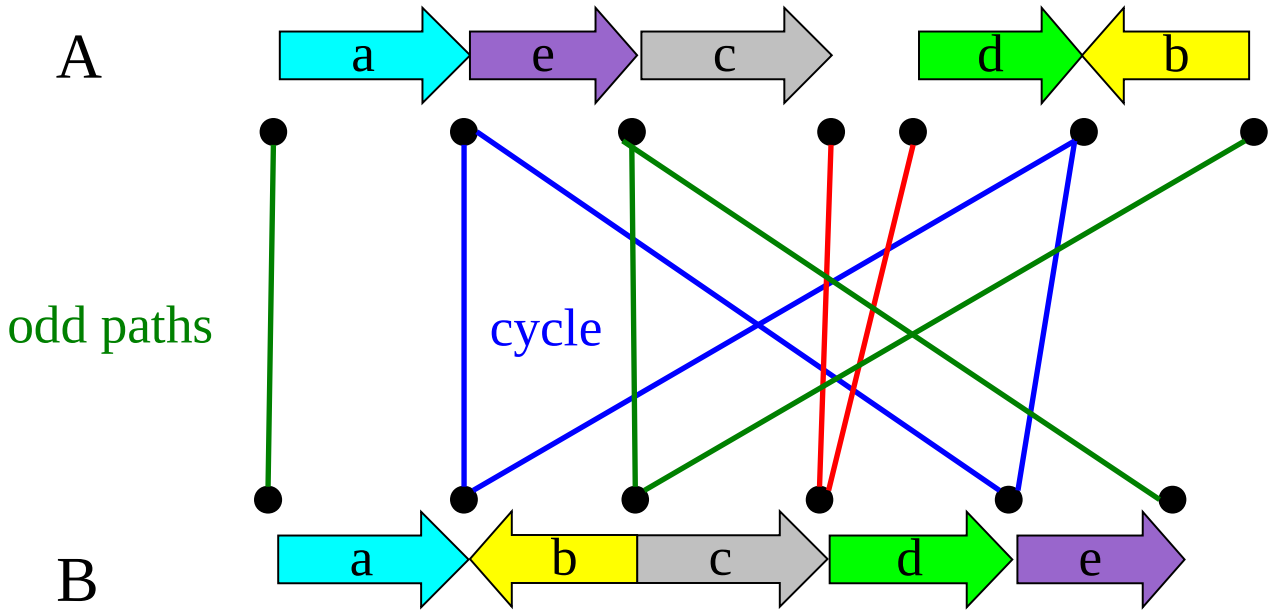
Adjacency Graph



- Bipartite graph, maximum degree equal to two: a collection of cycles and paths

(Bergeron et al. 2006)

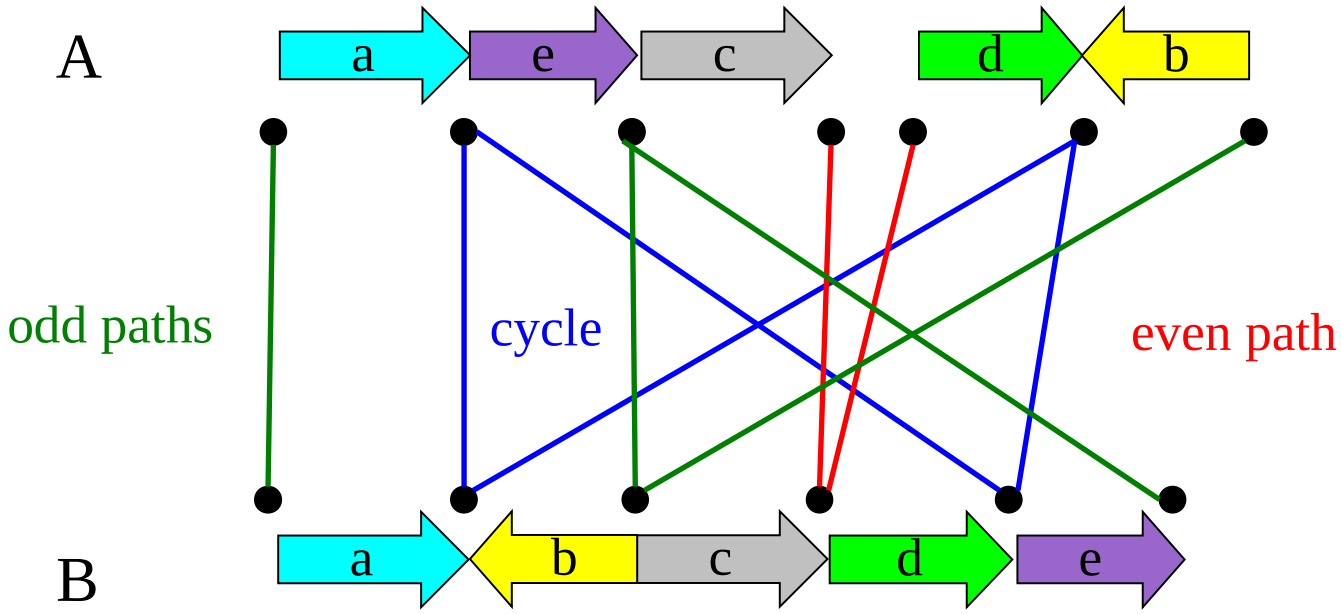
Adjacency Graph



- Bipartite graph, maximum degree equal to two: a collection of cycles and paths

(Bergeron et al. 2006)

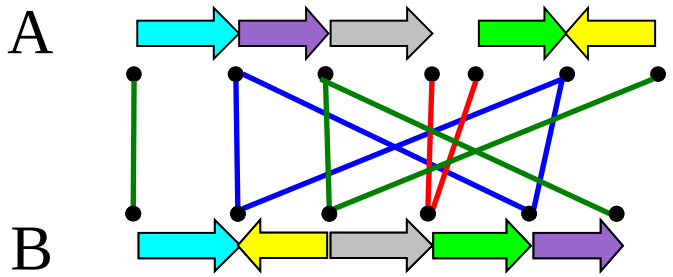
Adjacency Graph



- Bipartite graph, maximum degree equal to two: a collection of cycles and paths

(Bergeron et al. 2006)

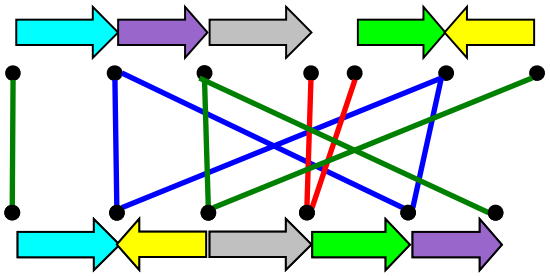
DCJ Sorting



(Bergeron et al. 2006)

DCJ Sorting

(Unsorted) A

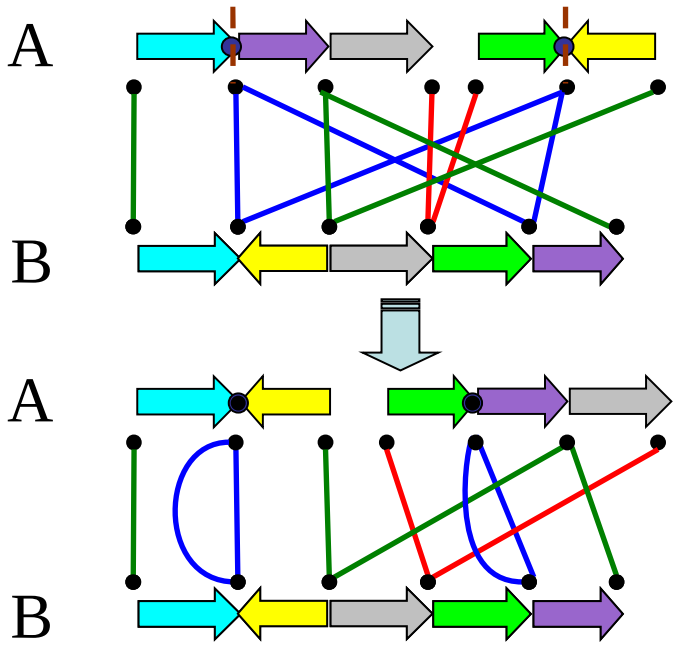


B

(Bergeron et al. 2006)

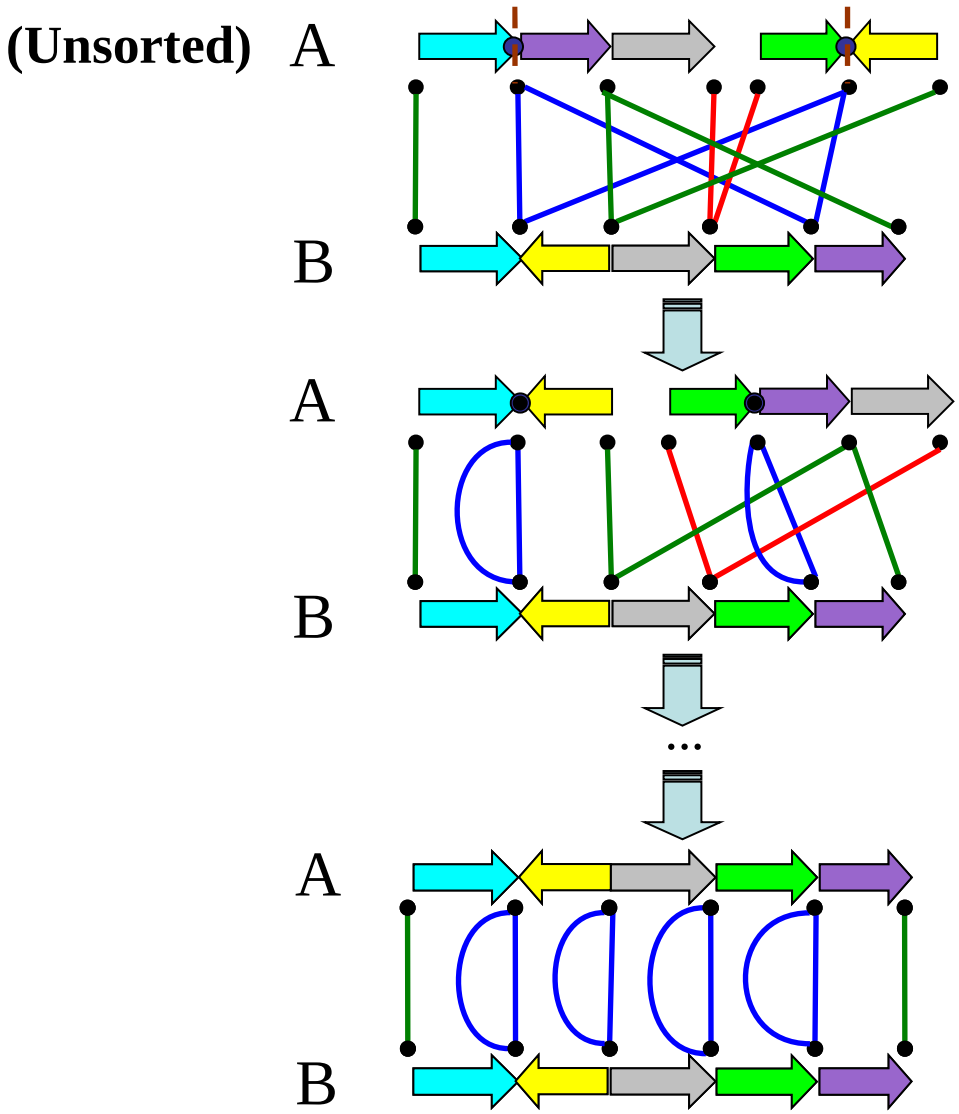
DCJ Sorting

(Unsorted)



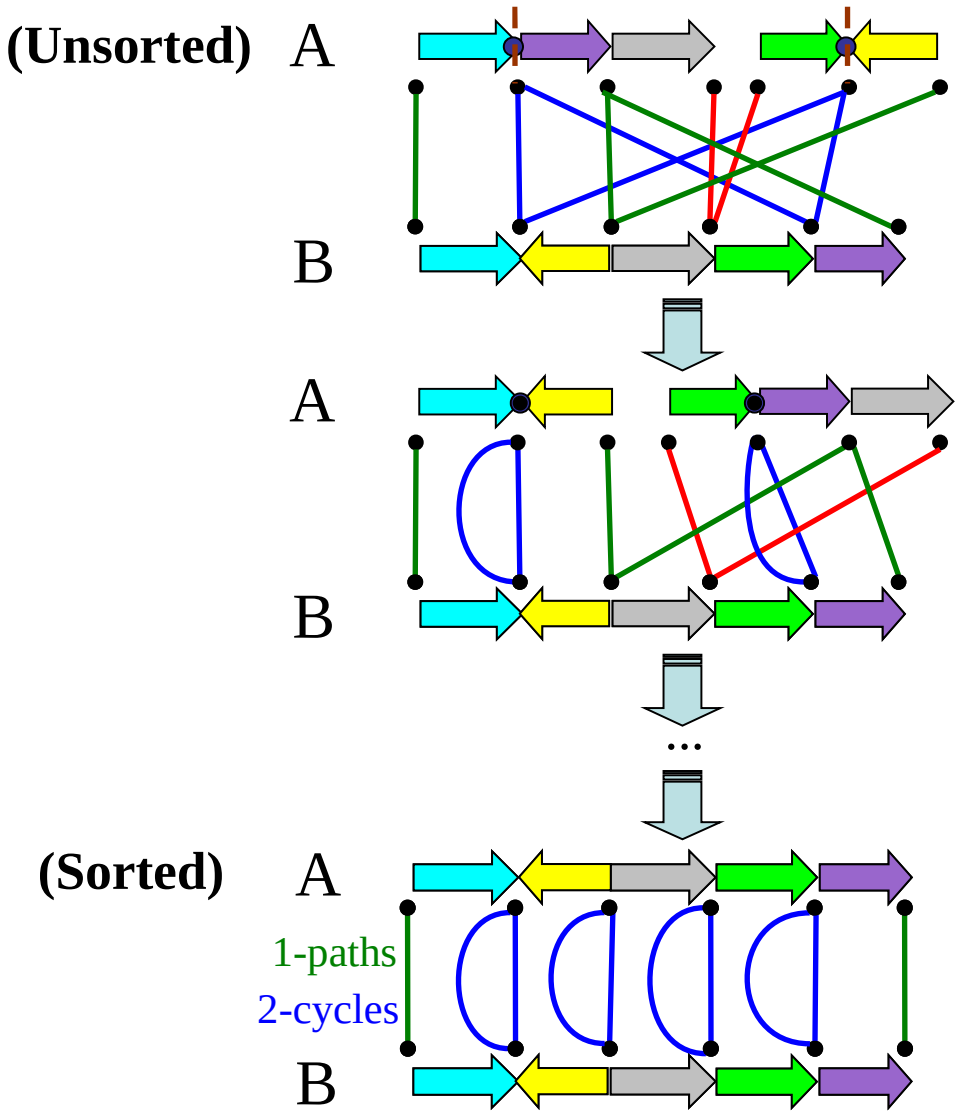
(Bergeron et al. 2006)

DCJ Sorting



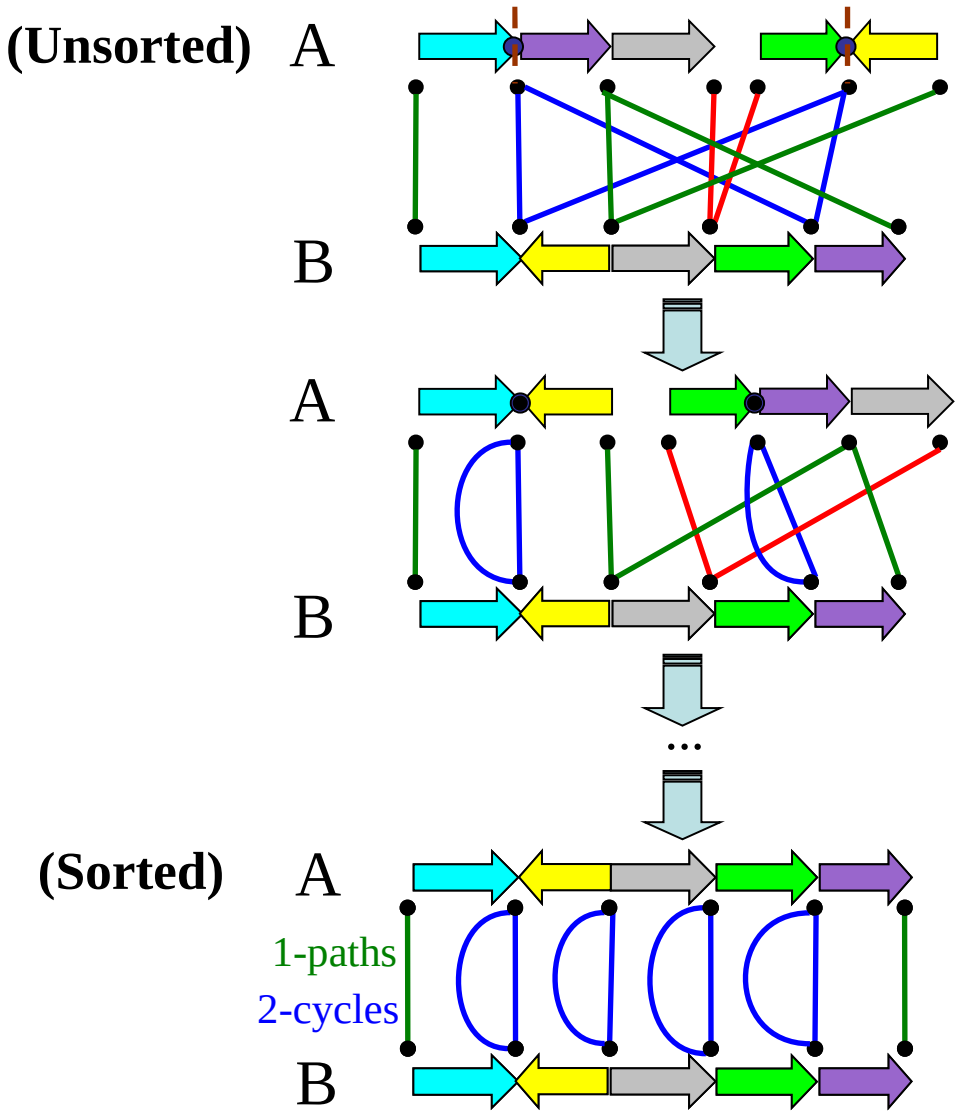
(Bergeron et al. 2006)

DCJ Sorting



(Bergeron et al. 2006)

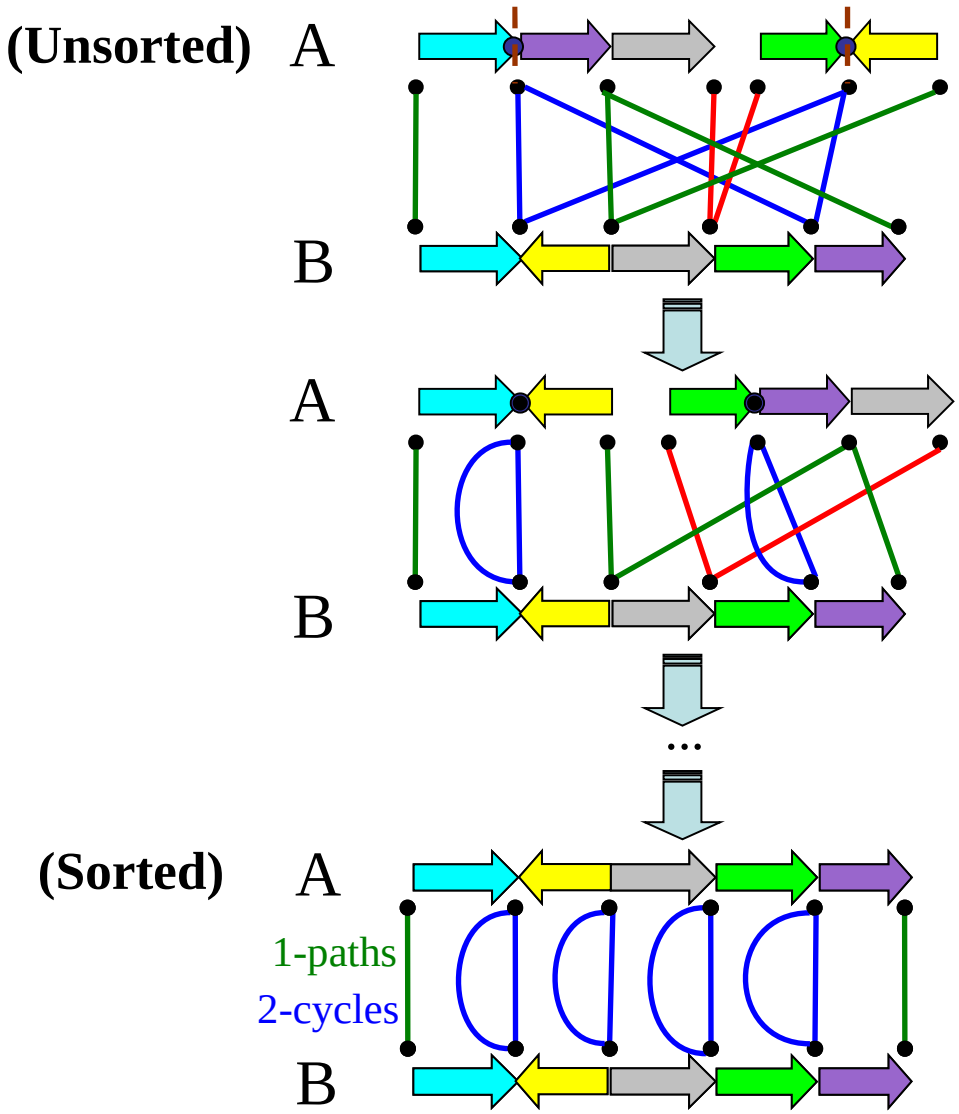
DCJ Sorting



Each optimal DCJ operation either increases the number of **cycles by one** or the number of **odd paths by two**

(Bergeron et al. 2006)

DCJ Sorting



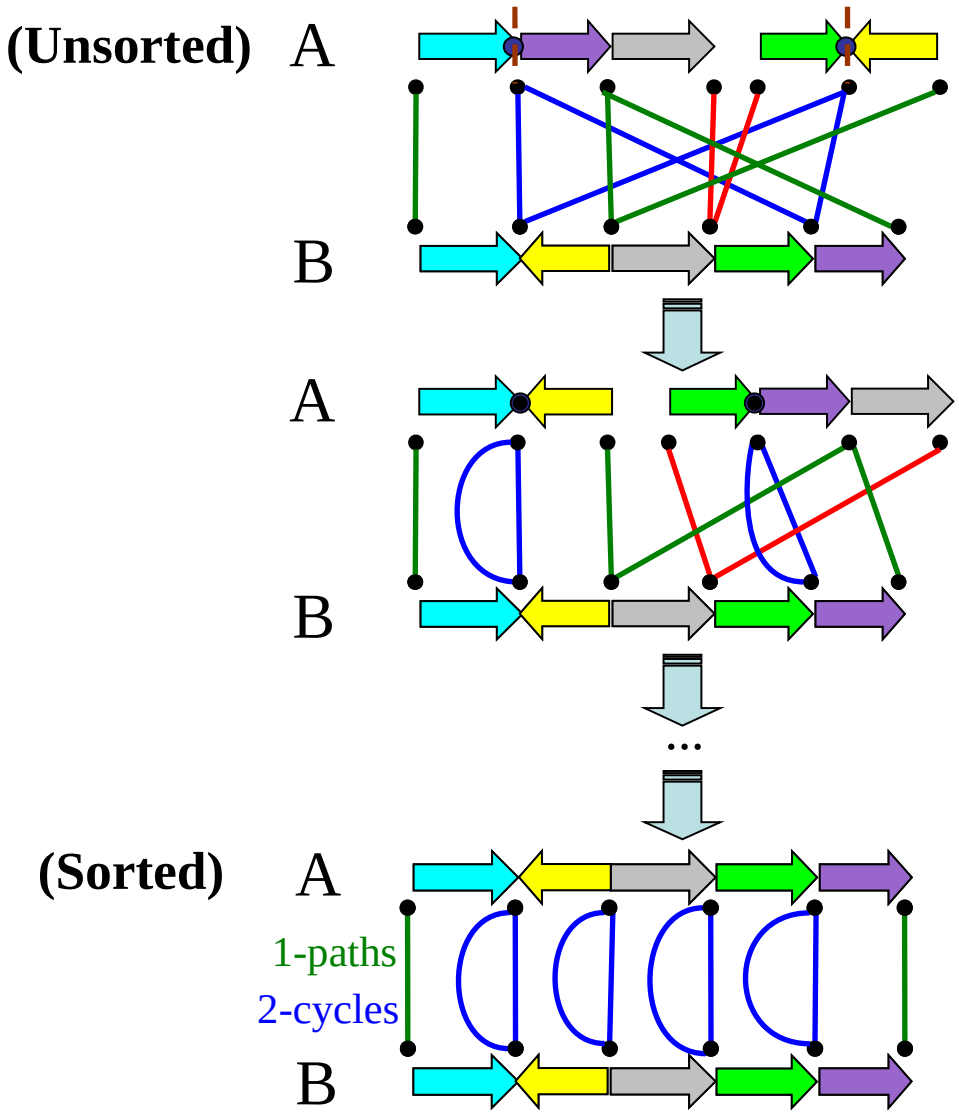
$$D = N - (C + B/2)$$

- N = number of genes
- C = number of cycles
- B = number of odd paths

Each optimal DCJ operation either increases the number of **cycles by one** or the number of **odd paths by two**

(Bergeron et al. 2006)

DCJ Sorting



$D = 5 - (1 + 2/2) = 3$

$$D = N - (C + B/2)$$

- N = number of genes
- C = number of cycles
- B = number of odd paths

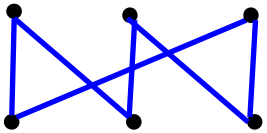
Each optimal DCJ operation either increases the number of **cycles by one** or the number of **odd paths by two**

(Bergeron et al. 2006)

Counting DCJ Sorting Scenarios

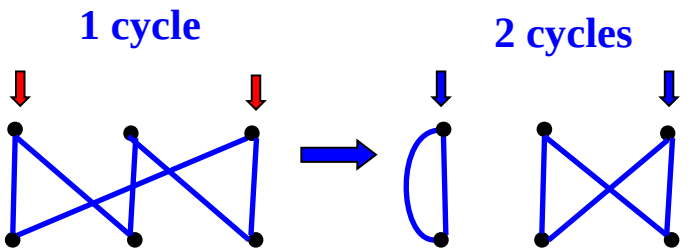
DCJ operations acting on vertices of genome A (top) belonging to the same cycle

1 cycle



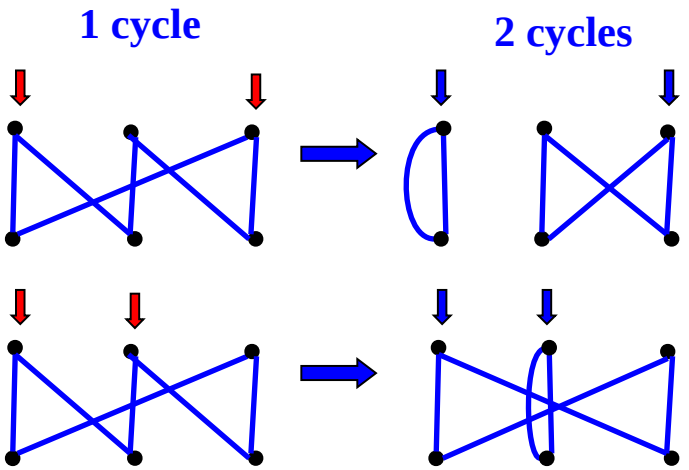
Counting DCJ Sorting Scenarios

DCJ operations acting on vertices of genome A (top) belonging to the same cycle



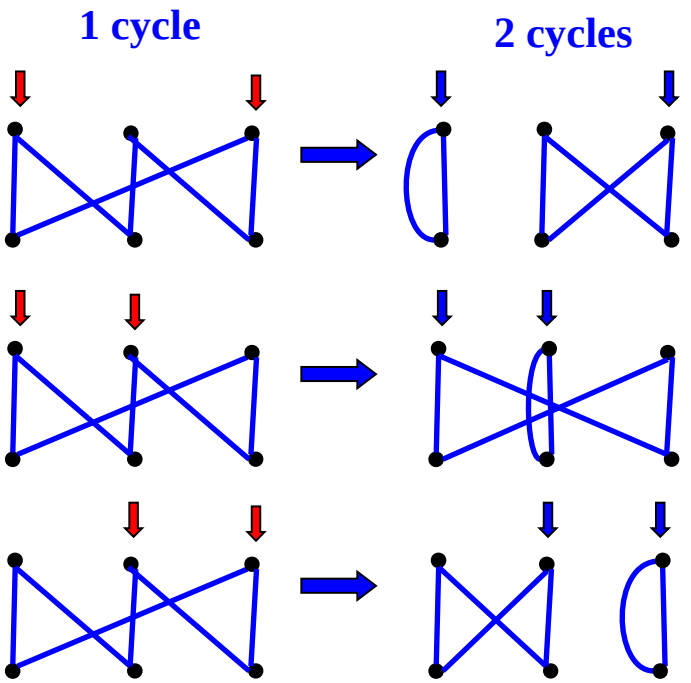
Counting DCJ Sorting Scenarios

DCJ operations acting on vertices of genome A (top) belonging to the same cycle



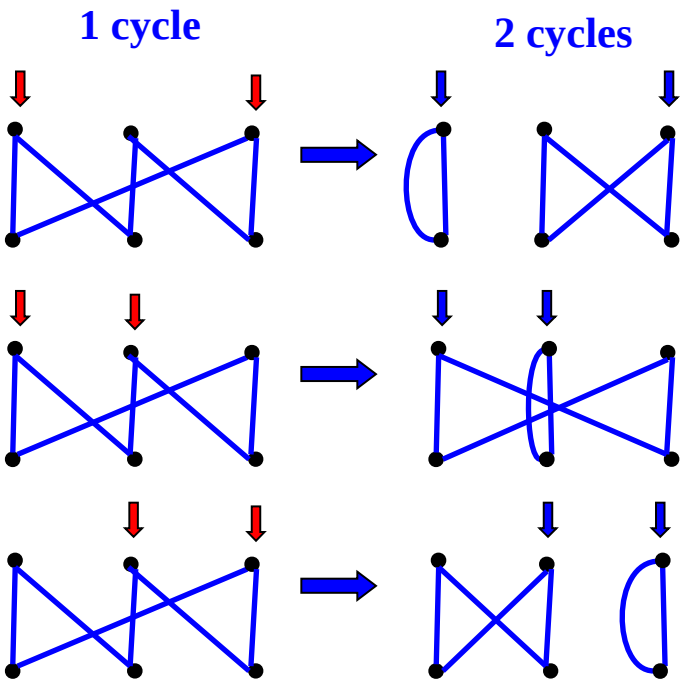
Counting DCJ Sorting Scenarios

DCJ operations acting on vertices of genome A (top) belonging to the same cycle



Counting DCJ Sorting Scenarios

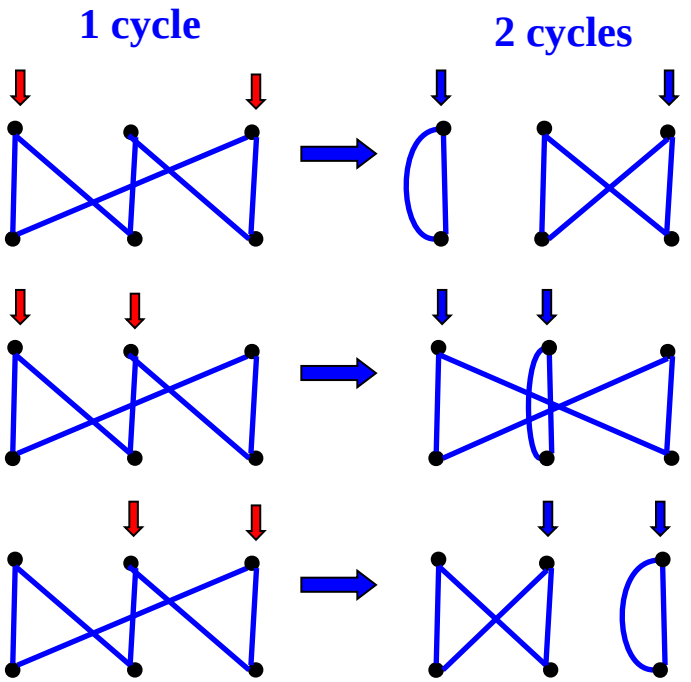
DCJ operations acting on vertices of genome A (top) belonging to the same cycle



For **each pair of vertices** there is **one optimal DCJ** operation

Counting DCJ Sorting Scenarios

DCJ operations acting on vertices of genome A (top) belonging to the same cycle



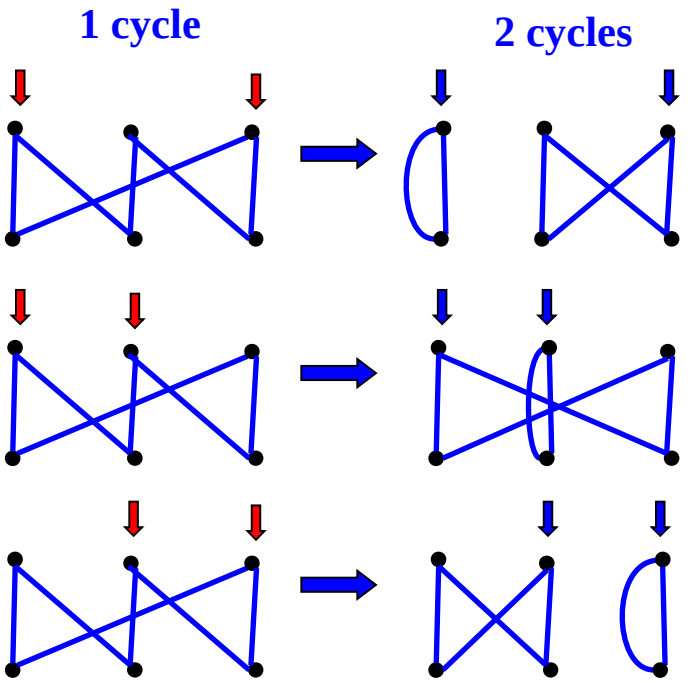
For **each pair of vertices** there is **one optimal DCJ** operation

A **(*k*+2)-cycle** in the adjacency graph can be sorted independently from the other components with $d = k/2$ operations in $(d+1)^{(d-1)}$ different ways

(k is even)

Counting DCJ Sorting Scenarios

DCJ operations acting on vertices of genome A (top) belonging to the same cycle



For **each pair of vertices** there is **one optimal DCJ** operation

Proven with a recurrence over the number of **vertices in genome A** (top):

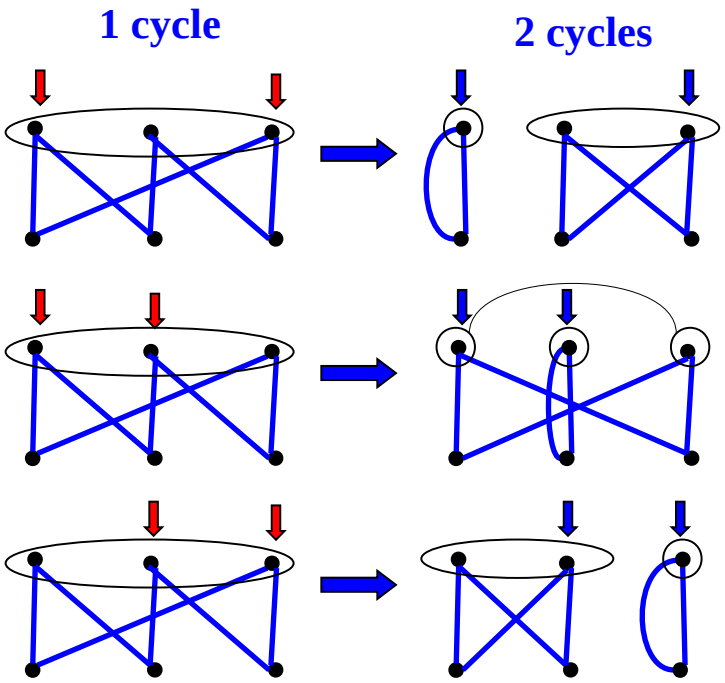
A cycle with ***v* vertices** can be split into a cycle with ***i* vertices** and a cycle with ***v-i* vertices** in ***v* different ways** (for ***i*** from 1 to ***v*/2**)

A **(*k*+2)-cycle** in the adjacency graph can be sorted independently from the other components with $d = k/2$ operations in $(d+1)^{(d-1)}$ different ways

(k is even)

Counting DCJ Sorting Scenarios

DCJ operations acting on vertices of genome A (top) belonging to the same cycle



For **each pair of vertices** there is **one optimal DCJ** operation

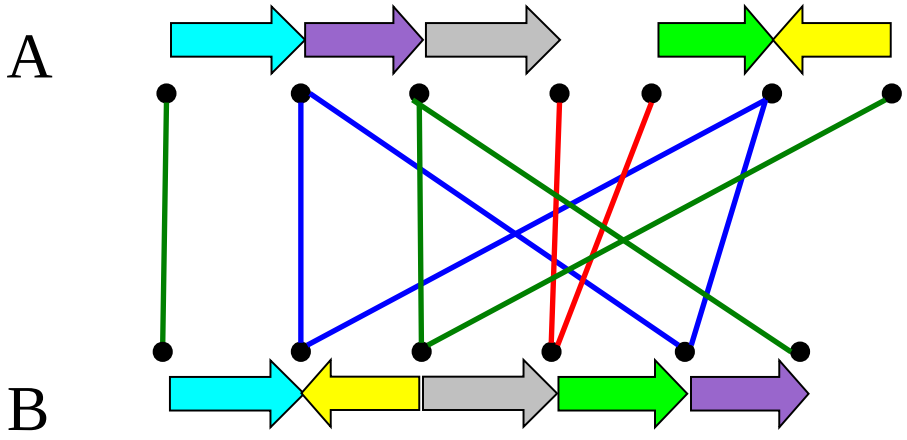
Proven with a recurrence over the number of **vertices in genome A** (top):

A cycle with ***v* vertices** can be split into a cycle with ***i* vertices** and a cycle with ***v-i* vertices** in ***v* different ways** (for ***i*** from 1 to ***v*/2**)

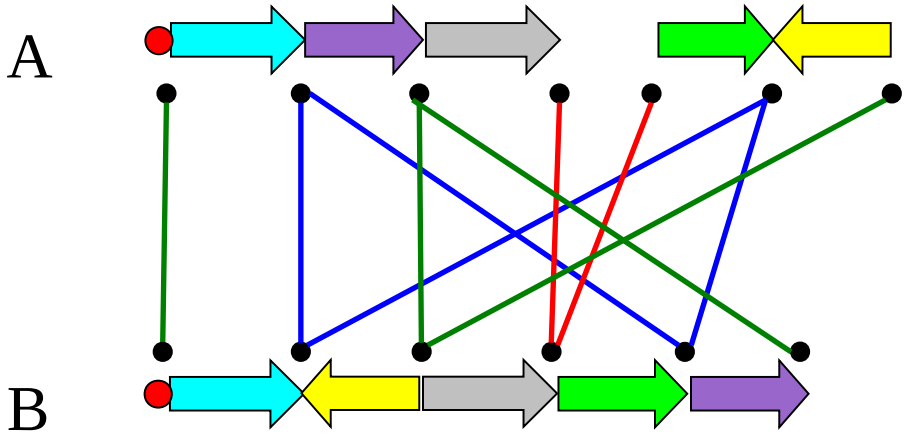
A **(*k*+2)-cycle** in the adjacency graph can be sorted independently from the other components with $d = k/2$ operations in $(d+1)^{(d-1)}$ different ways

(k is even)

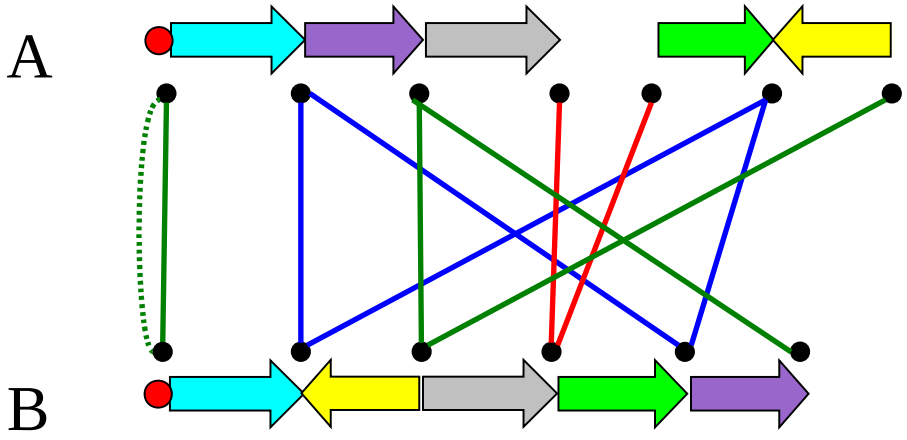
Counting DCJ Sorting Scenarios



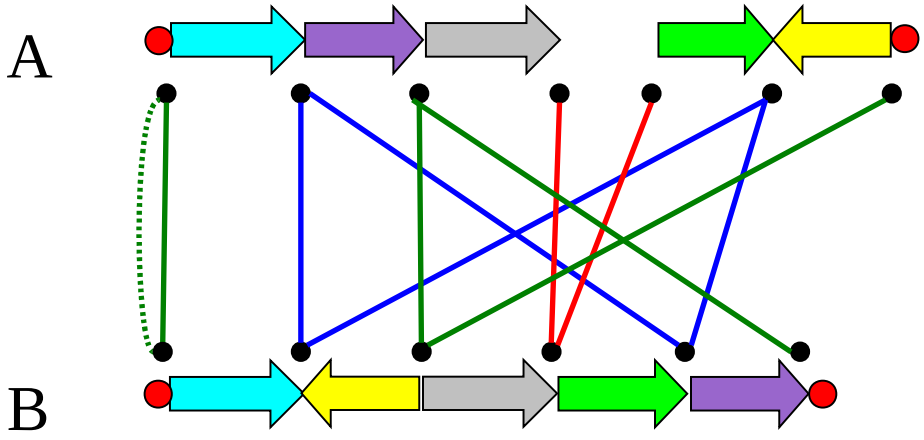
Counting DCJ Sorting Scenarios



Counting DCJ Sorting Scenarios

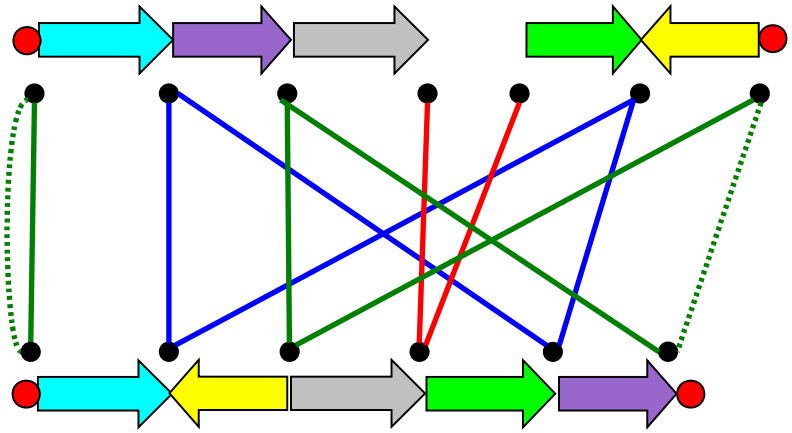


Counting DCJ Sorting Scenarios



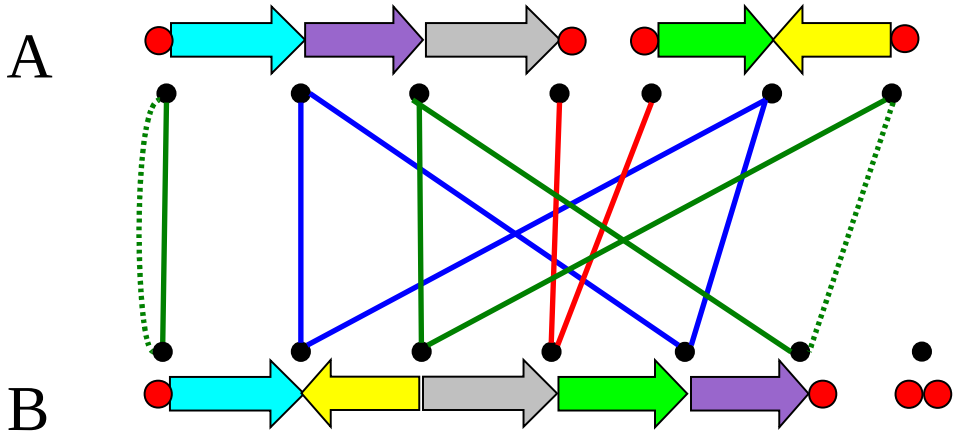
Counting DCJ Sorting Scenarios

A

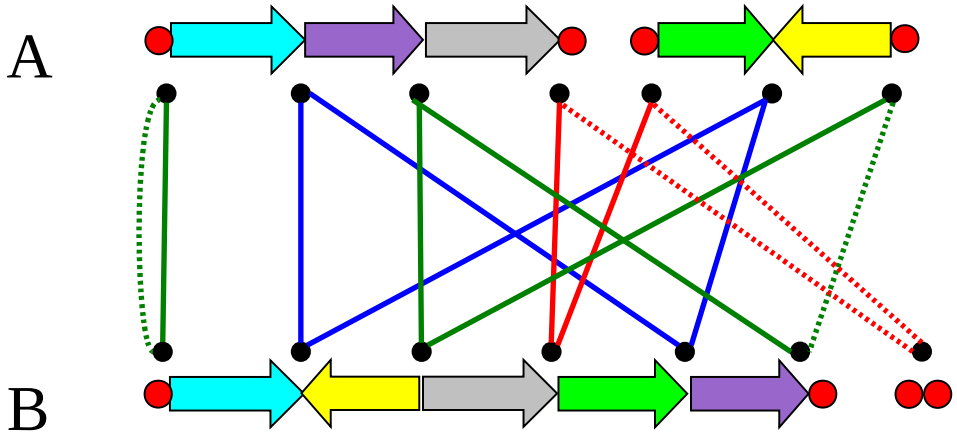


B

Counting DCJ Sorting Scenarios

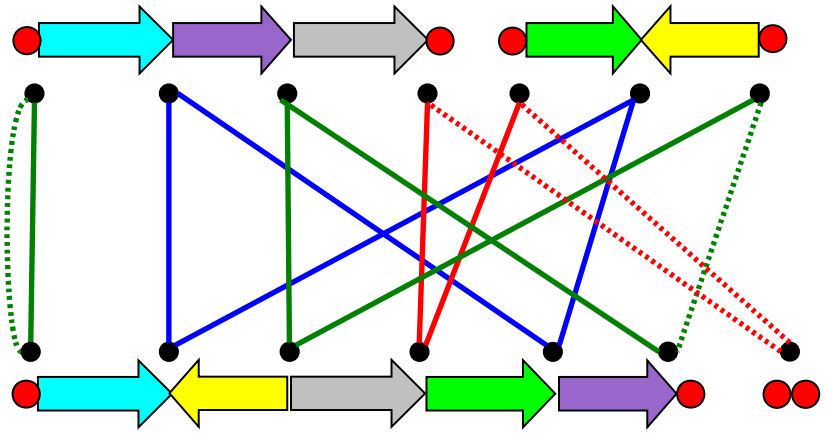


Counting DCJ Sorting Scenarios



Counting DCJ Sorting Scenarios

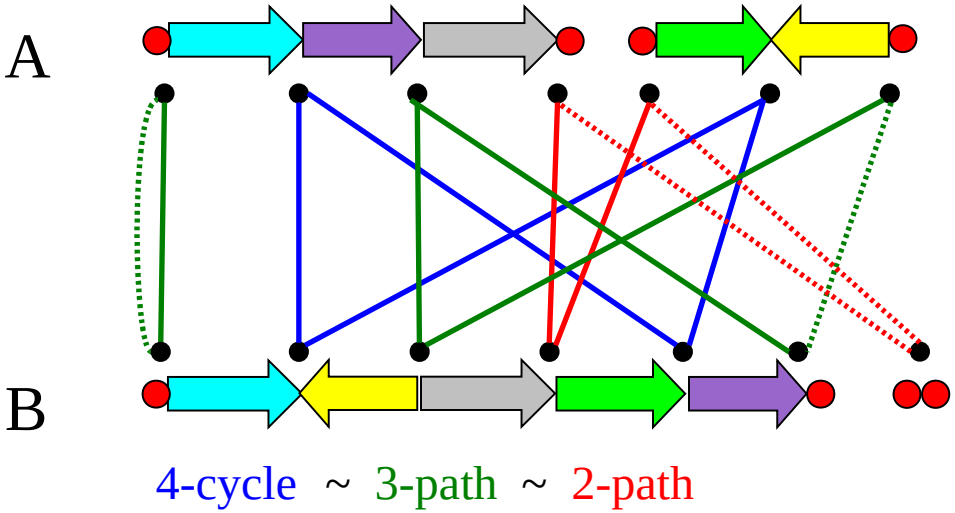
A



B

4-cycle ~ 3-path ~ 2-path

Counting DCJ Sorting Scenarios



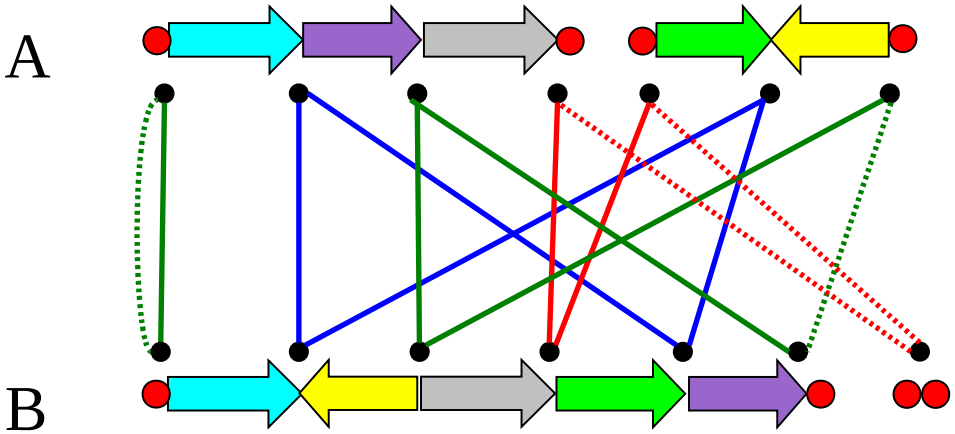
4-cycle ~ 3-path ~ 2-path

Sorting a (k)-path, or a (k+1)-path or a (k+2)-cycle :

- needs $d = k/2$ operations
- in $(d+1)^{(d-1)}$ different ways

(k is even)

Counting DCJ Sorting Scenarios



4-cycle ~ 3-path ~ 2-path

Sorting a (k)-path, or a (k+1)-path or a (k+2)-cycle :

- needs $d = k/2$ operations
- in $(d+1)^{(d-1)}$ different ways

(k is even)

All DCJ sorting scenarios obtained by combining subsequences sorting the n components independently:

$$\frac{(d_1+d_2+\cdots+d_n)!}{d_1!d_2!\ldots d_n!} \times (d_1+1)^{(d_1-1)} \times (d_2+1)^{(d_2-1)} \times \cdots \times (d_n+1)^{(d_n-1)}$$

Overview

- Introduction: the DCJ model
- Counting DCJ sorting scenarios
- Representing DCJ sorting scenarios
- Conclusions and perspectives