Algorithms in Genome Research

Pedro Feijão

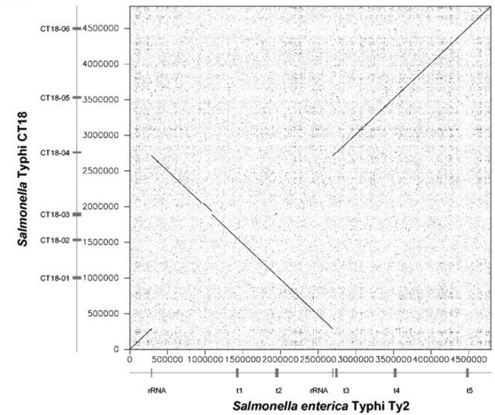
Winter 2013/14

pfeijao@cebitec.uni-bielefeld.de

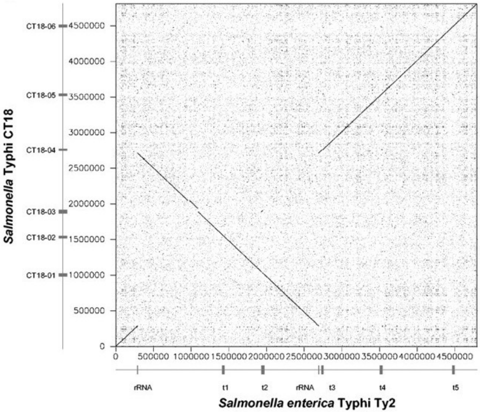
Before we start

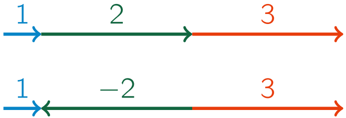
- Wiki Page:
  `http://wiki.techfak.uni-bielefeld.de/gi/Teaching`
- DiDy Workshop:
  `http://wiki.techfak.uni-bielefeld.de/didy`

Genome Rearrangements - Background

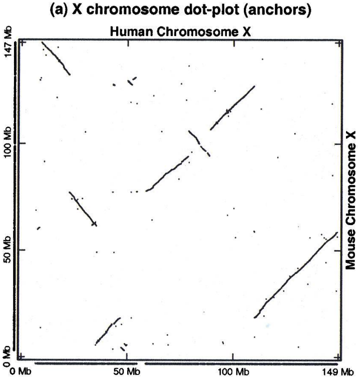


Genome Rearrangements - Background



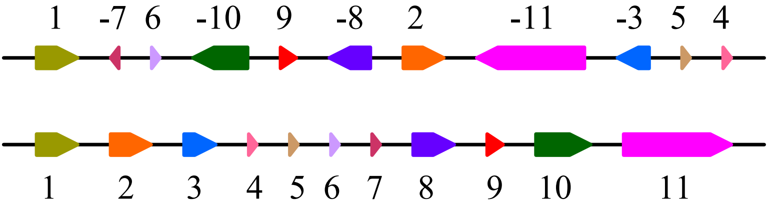


Human vs. Mouse

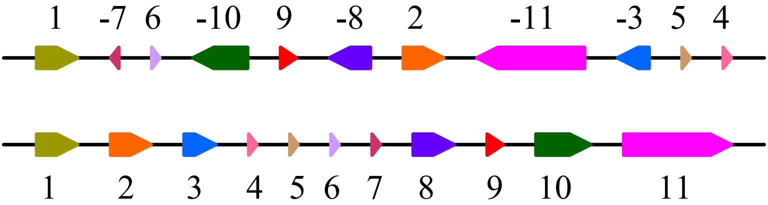


Pevzner, P.A. and Tesler, G. 2003. **Genome rearrangements in mammalian evolution: Lessons from human and mouse genomic sequences**. *Genome Res.* **13**: 13-26.

Human vs. Mouse

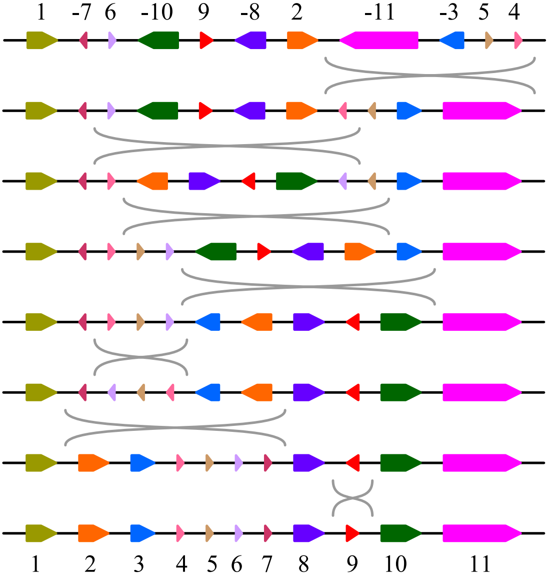


Human vs. Mouse



- How many rearrangements do we need to *transform* one genome into the other?

Human vs. Mouse



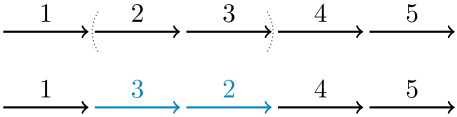
Genome Rearrangements

- **Genome rearrangements** are evolutionary events that *shuffle* the genome.
- Important questions:
    - What is the **minimum number** of rearrangement operations needed to transform one genome into another? (Distance)
    - Can we find a **rearrangement scenario** with this minimum number of operations? (Sorting)

Genome Rearrangements

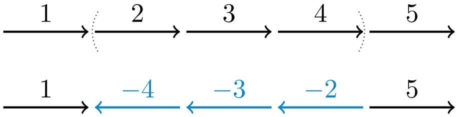
- **Genome rearrangements** are evolutionary events that *shuffle* the genome.
- Important questions:
    - What is the **minimum number** of rearrangement operations needed to transform one genome into another? (Distance)
    - Can we find a **rearrangement scenario** with this minimum number of operations? (Sorting)
- Several types of **rearrangement operations** were studied:

Genome Rearrangements



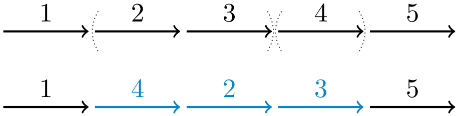
Unsigned Reversal/Inversion

Genome Rearrangements



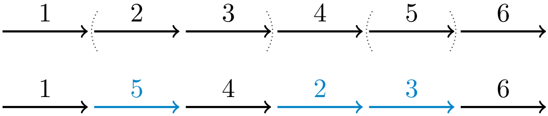
Signed Reversal/Inversion

Genome Rearrangements



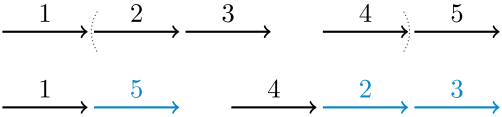
Transposition

Genome Rearrangements



Block Interchange

Genome Rearrangements



Translocation (*multichromosomal* operation)

Genome Rearrangement Models

- Several models were proposed, allowing only one operation or combining two or more.
- Usually polinomially solvable, notable exceptions: Unsigned reversal and Transposition (NP-hard)

Inversion Models

- Since 1990, beginning with Sankoff in 1992, many papers have been devoted to the subject of **inversion distance**.

Inversion Models

- Since 1990, beginning with Sankoff in 1992, many papers have been devoted to the subject of **inversion distance**.
- The *unsigned inversion* distance is NP-hard (Caprara 1997)

Inversion Models

- Since 1990, beginning with Sankoff in 1992, many papers have been devoted to the subject of **inversion distance**.
- The *unsigned inversion* distance is NP-hard (Caprara 1997)
- The *signed inversion* was solved polynomially by Hannenhalli and Pevzner in 1995. It is usually called **HP model**.

Inversion Models

- Since 1990, beginning with Sankoff in 1992, many papers have been devoted to the subject of **inversion distance**.
- The *unsigned inversion* distance is NP-hard (Caprara 1997)
- The *signed inversion* was solved polynomially by Hannenhalli and Pevzner in 1995. It is usually called **HP model**.
- The HP model was later improved and simplified in a series of articles. Here we will present elements of the original theory, also with contributions from Bergeron (2001) and also Bergeron,Mixtacki and Stoye (2005).

Definitions

- A **signed permutation** is a permutations on the set $\{0, 1, \ldots, n\}$ in which every element has a *sign*. In our case the permutations always start with 0 and end with n.

Definitions

- A **signed permutation** is a permutations on the set $\{0, 1, \dots, n\}$ in which every element has a *sign*. In our case the permutations always start with 0 and end with n. *For example*:

  $$\pi_1 = (0 \quad -2 \quad -1 \quad 4 \quad 3 \quad 5 \quad -8 \quad 6 \quad 7 \quad 9)$$

Definitions

- A **signed permutation** is a permutations on the set $\{0, 1, \ldots, n\}$ in which every element has a *sign*. In our case the permutations always start with 0 and end with n. *For example*:

$$\pi_1 = (0 \quad -2 \quad -1 \quad\quad 4 \quad\quad 3 \quad\quad 5 \quad -8 \quad\quad 6 \quad\quad 7 \quad\quad 9)$$

- A **point** $p \cdot q$ is a pair of consecutive elements in the permutation. In the above example, $0 \cdot -2$ and $-2 \cdot -1$ are the first two points of π_1.

Definitions

- A **signed permutation** is a permutations on the set $\{0, 1, \ldots, n\}$ in which every element has a *sign*. In our case the permutations always start with 0 and end with n. *For example*:

$$\pi_1 = (0 \quad -2 \quad -1 \quad 4 \quad 3 \quad 5 \quad -8 \quad 6 \quad 7 \quad 9)$$

- A **point** $p \cdot q$ is a pair of consecutive elements in the permutation. In the above example, $0 \cdot -2$ and $-2 \cdot -1$ are the first two points of π_1.
- When a point is in the form $i \cdot (i+1)$ or $-(i+1) \cdot -i$ it is called an **(conserved) adjacency**. Otherwise, it is a **breakpoint**.

Breakpoints

$$\pi_1 = (0 \quad -2 \quad -1 \quad 4 \quad 3 \quad 5 \quad -8 \quad 6 \quad 7 \quad 9)$$

- In this permutation, there are *two* adjacencies, $-2 \cdot -1$ and $6 \cdot 7$, and *seven* breakpoints.

Breakpoints

$$\pi_1 = (0 \quad -2 \quad -1 \quad 4 \quad 3 \quad 5 \quad -8 \quad 6 \quad 7 \quad 9)$$

- In this permutation, there are *two* adjacencies, $-2 \cdot -1$ and $6 \cdot 7$, and *seven* breakpoints.
- The **Breakpoint Distance** is the number of breakpoints in a permutation, that is, distance from the **identity**:

$$\mathrm{Id} = (0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9)$$

Breakpoints

$$\pi_1 = (0 \quad -2 \quad -1 \quad 4 \quad 3 \quad 5 \quad -8 \quad 6 \quad 7 \quad 9)$$

- In this permutation, there are *two* adjacencies, $-2 \cdot -1$ and $6 \cdot 7$, and *seven* breakpoints.
- The **Breakpoint Distance** is the number of breakpoints in a permutation, that is, distance from the **identity**:

$$\mathrm{Id} = (0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9)$$

- It is one the simplest measure of dissimilarity for genome rearrangements. *Notation*: $d_{\mathsf{BP}}(\pi_1) = 7$.

Breakpoints

$$\pi_1 = (0 \quad -2 \quad -1 \quad 4 \quad 3 \quad 5 \quad -8 \quad 6 \quad 7 \quad 9)$$

- In this permutation, there are *two* adjacencies, $-2 \cdot -1$ and $6 \cdot 7$, and *seven* breakpoints.
- The **Breakpoint Distance** is the number of breakpoints in a permutation, that is, distance from the **identity**:

$$\mathrm{Id} = (0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9)$$

- It is one the simplest measure of dissimilarity for genome rearrangements. *Notation*: $d_{\mathsf{BP}}(\pi_1) = 7$.

Breakpoints

$$\pi_1 = (0 \quad -2 \quad -1 \quad 4 \quad 3 \quad 5 \quad -8 \quad 6 \quad 7 \quad 9)$$

- In this permutation, there are *two* adjacencies, $-2 \cdot -1$ and $6 \cdot 7$, and *seven* breakpoints.
- The **Breakpoint Distance** is the number of breakpoints in a permutation, that is, distance from the **identity**:

$$\mathrm{Id} = (0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9)$$

- It is one the simplest measure of dissimilarity for genome rearrangements. *Notation*: $d_{\mathsf{BP}}(\pi_1) = 7$.

For instance, the permutation

$$\pi_2 = (0 \quad -4 \quad -3 \quad -2 \quad -1 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9)$$

has 2 breakpoints, which means that π_2 is *closer* to the identity than π_1.

Inversions

- An **inversion** of a permutation interval reverts the *order* and *sign* of all elements of the interval.

$$\pi_1 = (0 \quad -2 \quad \underline{-1 \quad 4 \quad 3 \quad 5} \quad -8 \quad 6 \quad 7 \quad 9)$$

Inversions

- An **inversion** of a permutation interval reverts the *order* and *sign* of all elements of the interval.

$$\begin{array}{cccccccccccc} \pi_1 & = & (0 & -2 & \underline{-1 \quad 4 \quad 3 \quad 5} & -8 & 6 & 7 & 9) \\ \pi_1' & = & (0 & -2 & -5 \quad -3 \quad -4 \quad 1 & -8 & 6 & 7 & 9) \end{array}$$

Inversions

- An **inversion** of a permutation interval reverts the *order* and *sign* of all elements of the interval.

$$\begin{array}{lcllllllllll}
\pi_1 & = & (0 & -2 & \underline{-1 \quad 4 \quad 3 \quad 5} & -8 & 6 & 7 & 9) \\
\pi_1' & = & (0 & -2 & -5 \quad -3 \quad -4 \quad 1 & -8 & 6 & 7 & 9)
\end{array}$$

- The **inversion distance** is the minimum number of inversions needed to transform one permutation into another (usually the other permutation is the identity). Notation: $d_R(\pi_1)$.
- Finding such a scenario of inversions is called **sorting by inversions**.
  - *Distance* vs. *Sorting*

BP vs. Inversions

- A inversion changes the number of breakpoints by at most 2.
- This gives a simple *lower bound* for the inversion distance:

$$d_R(\pi_1) \geq \frac{d_{\mathrm{BP}}(\pi_1)}{2}$$

- Using BP for lower bound is an useful *first approach* in many models.

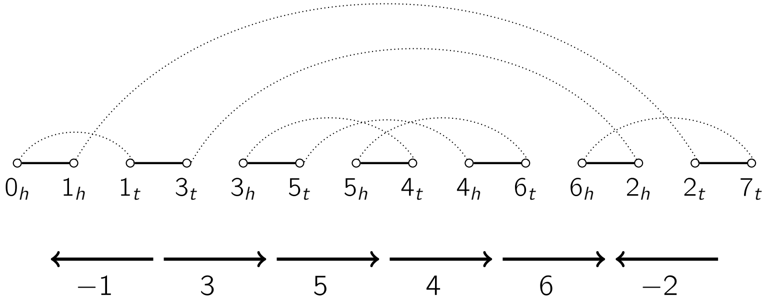
Breakpoint Graph - Genomes as Graphs

Breakpoint Graph - Genomes as Graphs

- The BP graph of a is a very useful structure for studying rearrangement problems. Notation $BP(\pi)$.
- **Vertices** are the gene extremities (tail and head).
- **Black edges** between consecutive gene extremities (reality edges).
- **Grey edges** between consecutive gene extremities of the identity (desire edges).

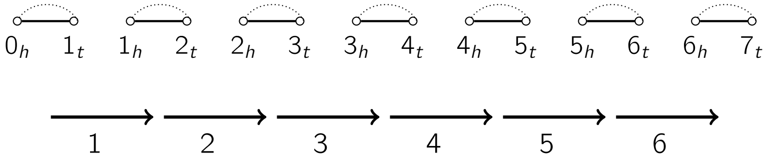
Breakpoint Graph - Genomes as Graphs

- The BP graph of a is a very useful structure for studying rearrangement problems. Notation $BP(\pi)$.
- **Vertices** are the gene extremities (tail and head).
- **Black edges** between consecutive gene extremities (reality edges).
- **Grey edges** between consecutive gene extremities of the identity (desire edges).



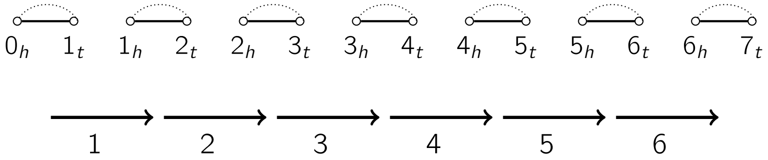
Breakpoint Graph

- When the input genome is the identity, the BP graph is composed of n **trivial cycles**.



Breakpoint Graph

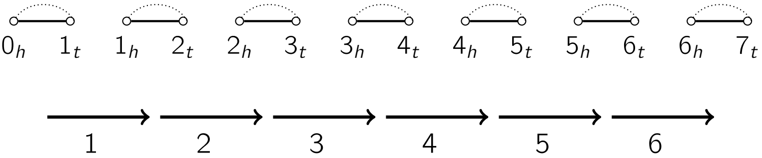
- When the input genome is the identity, the BP graph is composed of *n* **trivial cycles**.



- Sorting is equivalent to **increasing the cycles of the BP graph**.

Breakpoint Graph

- When the input genome is the identity, the BP graph is composed of *n* **trivial cycles**.



- Sorting is equivalent to **increasing the cycles of the BP graph**.
- What happens in the BP graph when an inversion is applied?

Breakpoint Graph - Lower Bound

- An inversion changes the number of cycles of the BP graph at most by 1.

Breakpoint Graph - Lower Bound

- An inversion changes the number of cycles of the BP graph at most by 1.
- Again, we have a **lower bound** for the inversion distance:

$$d_R(\pi) \geq N - C$$

where C is the *number of cycles* in the BP graph of π.

Breakpoint Graph - Lower Bound

- An inversion changes the number of cycles of the BP graph at most by 1.
- Again, we have a **lower bound** for the inversion distance:

  $$d_R(\pi) \geq N - C$$

  where C is the *number of cycles* in the BP graph of π.
- This bound is **very tight**, that is, usually it is exactly the inversion distance.

Breakpoint Graph - Lower Bound

- An inversion changes the number of cycles of the BP graph at most by 1.
- Again, we have a **lower bound** for the inversion distance:

  $$d_R(\pi) \geq N - C$$

  where C is the *number of cycles* in the BP graph of π.
- This bound is **very tight**, that is, usually it is exactly the inversion distance.
- When is this bound not *exactly* the distance?

Breakpoint Graph - Lower Bound

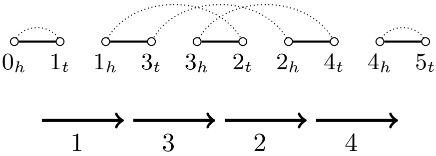
- An inversion changes the number of cycles of the BP graph at most by 1.
- Again, we have a **lower bound** for the inversion distance:

  $$d_R(\pi) \geq N - C$$

  where C is the *number of cycles* in the BP graph of π.
- This bound is **very tight**, that is, usually it is exactly the inversion distance.
- When is this bound not *exactly* the distance?
  - When it is not possible to increase the cycles of BP with an inversion.
  - That occurs in the presence of **unoriented components**.

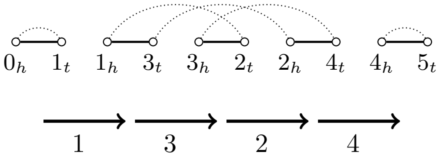
Unoriented components

- In the example below, there is no inversion that increases the number of cycles.



Unoriented components

- In the example below, there is no inversion that increases the number of cycles.



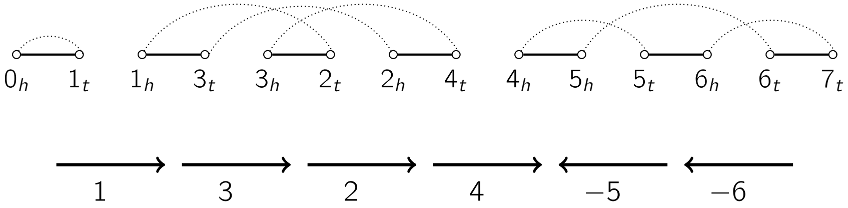
- The lower bound is $N - C = 5 - 3 = 2$, but the real distance is 3, because one extra reversal is needed to *orient* the unoriented cycle in the BP graph.
- So, let's define oriented/unoriented components.

BP Graph Components

- Two black edges in a same cycle are **convergent** if, when traversing the cycle both edges induce the *same direction*. Otherwise, they are **divergent**.

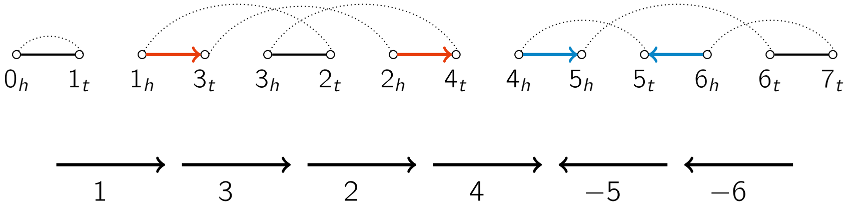
BP Graph Components

- Two black edges in a same cycle are **convergent** if, when traversing the cycle both edges induce the *same direction*. Otherwise, they are **divergent**.



BP Graph Components

- Two black edges in a same cycle are **convergent** if, when traversing the cycle both edges induce the *same direction*. Otherwise, they are **divergent**.

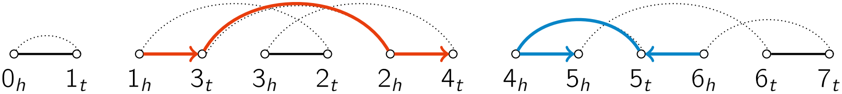


BP Graph Components

- A grey edge is **oriented** if its two incident black edges are *divergent*, otherwise the edge is **unoriented**.

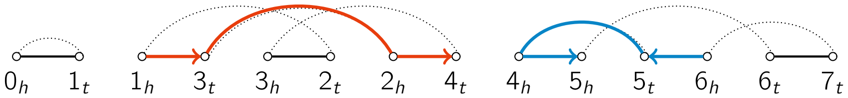
BP Graph Components

- A grey edge is **oriented** if its two incident black edges are *divergent*, otherwise the edge is **unoriented**.



BP Graph Components

- A grey edge is **oriented** if its two incident black edges are *divergent*, otherwise the edge is **unoriented**.



- Equivalently, A grey edge is **oriented** if it "contains" an odd number of vertices, and **unoriented** otherwise (even number of vertices).

BP Graph Components

- A cycle is **oriented** if it contains *at least one* oriented edge. Otherwise, it is **unoriented**.

BP Graph Components

- A cycle is **oriented** if it contains *at least one* oriented edge. Otherwise, it is **unoriented**.

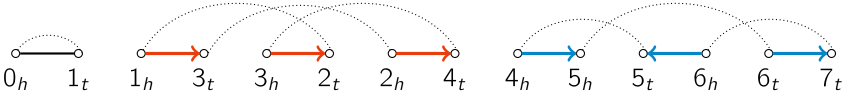


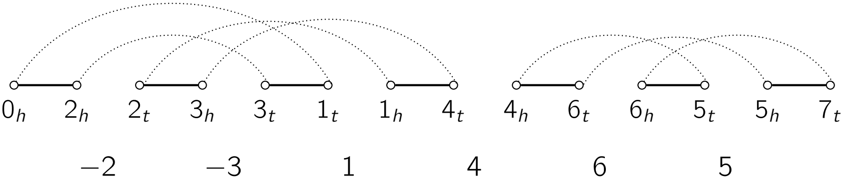
Figure : Example of unoriented and oriented cycles.

BP Graph Components

- Two cycles are **connected** if they have overlapping edges.
- A **component** is a subset of connected cycles.

BP Graph Components

- Two cycles are **connected** if they have overlapping edges.
- A **component** is a subset of connected cycles.



- An **oriented (good) component** has at least one oriented cycle, otherwise it is a **unoriented (bad) component**.

Sorting good components

Theorem (Hannenhalli-Pevzer, 95)

If the graph $BP(\pi)$ *has only* **oriented components**, *then*

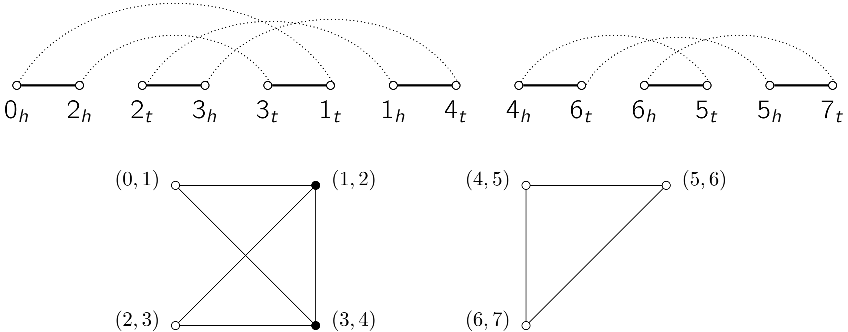
$$d_R(\pi) = N - C$$

where N *is the number of elements of* π *and* C *is the number of cycles of* $BP(\pi)$.

- When there are only oriented components, there is always at least one inversion that increases the number of cycles of $BP(\pi)$ and *does not create any unoriented component*.
- These are called **safe inversions**.

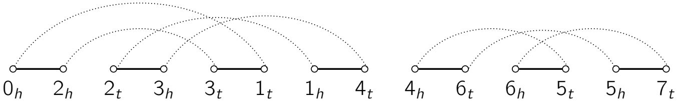
Finding safe inversions - Definitions

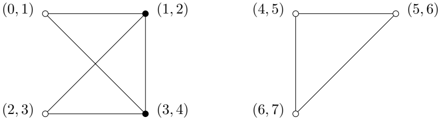
- The **overlap graph** $O(\pi)$ is a graph where:
  - Vertices are the grey edges of $BP(\pi)$. If the edge is oriented, the vertex is black, otherwise is white.
  - When two grey edges overlap, there is an edge between the corresponding vertices.



BP Graph vs Overlap Graph

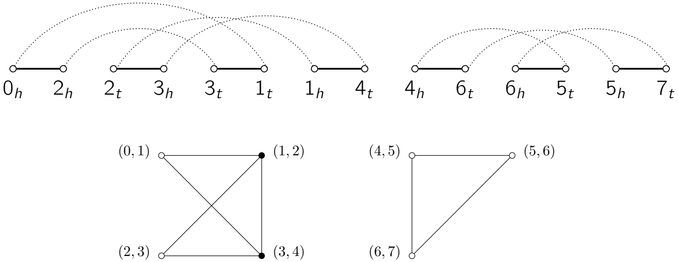
BP Graph	**Overlap Graph**
Component	Connected component





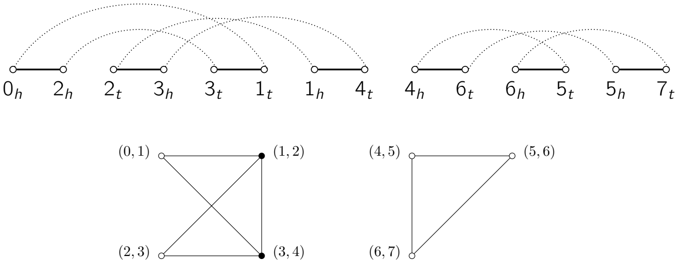
BP Graph vs Overlap Graph

BP Graph	**Overlap Graph**
Component	Connected component
Oriented edge	Black vertex, *odd degree*



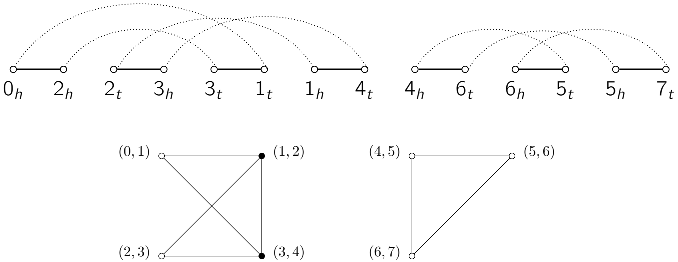
BP Graph vs Overlap Graph

BP Graph	**Overlap Graph**
Component	Connected component
Oriented edge	Black vertex, *odd degree*
Unoriented edge	White vertex, *even degree*



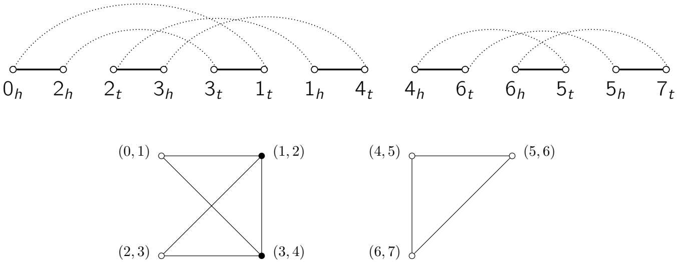
BP Graph vs Overlap Graph

BP Graph	**Overlap Graph**
Component	Connected component
Oriented edge	Black vertex, *odd degree*
Unoriented edge	White vertex, *even degree*
Oriented component	Component with at least 1 black vertex

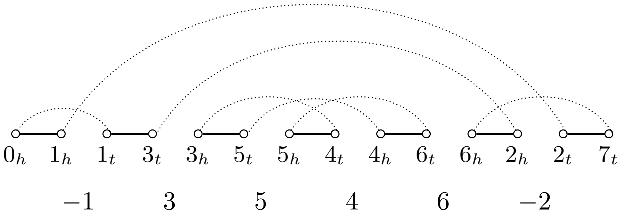


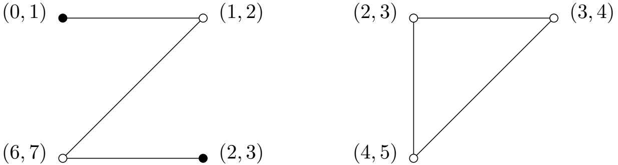
BP Graph vs Overlap Graph

BP Graph	**Overlap Graph**
Component	Connected component
Oriented edge	Black vertex, *odd degree*
Unoriented edge	White vertex, *even degree*
Oriented component	Component with at least 1 black vertex
Unoriented component	Component with only white vertices



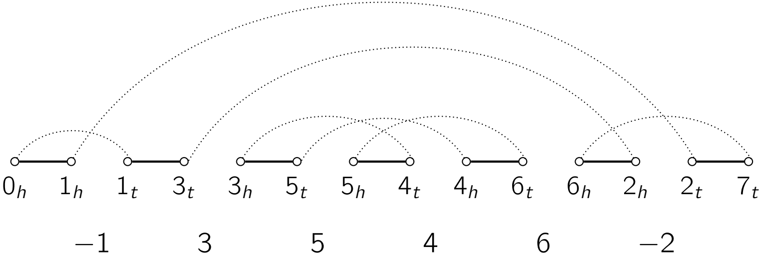
Another Example





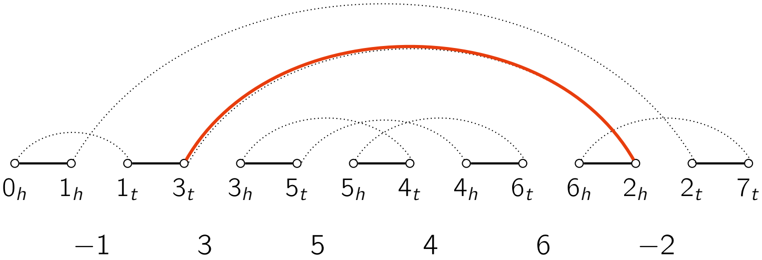
Inducing Inversions

- A inversion **induced** by an oriented BP edge reverses the elements that are *completely* contained in the edge.



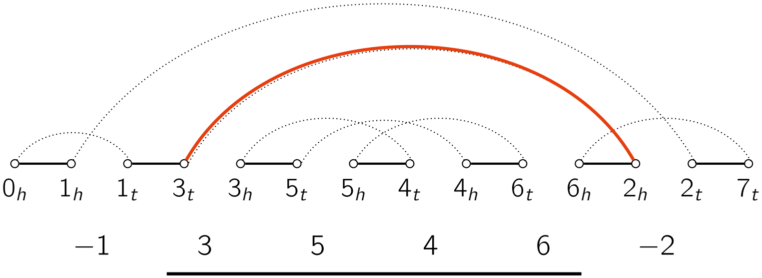
Inducing Inversions

- A inversion **induced** by an oriented BP edge reverses the elements that are *completely* contained in the edge.



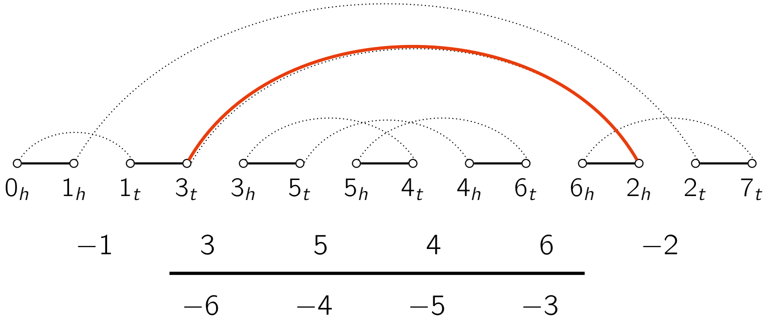
Inducing Inversions

- A inversion **induced** by an oriented BP edge reverses the elements that are *completely* contained in the edge.



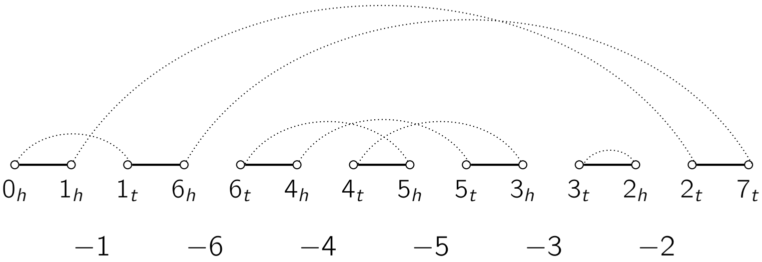
Inducing Inversions

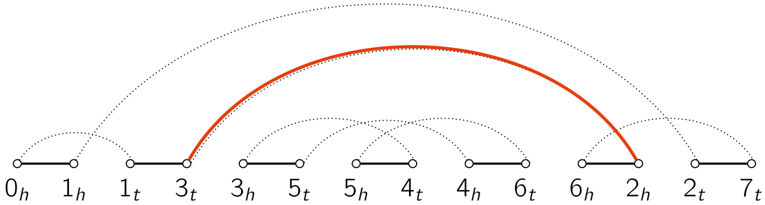
- A inversion **induced** by an oriented BP edge reverses the elements that are *completely* contained in the edge.



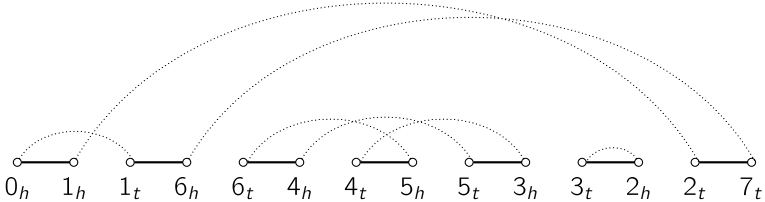
Inducing Inversions

- After applying the reversal, the adjacency $-3 \cdot -2$ is created, and the number of cycles increases by 1.





$$-1 \quad \underline{3 \quad 5 \quad 4 \quad 6} \quad -2$$

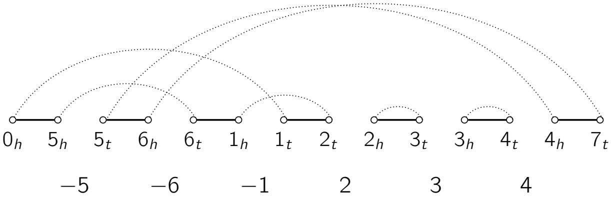


$$-1 \quad -6 \quad -4 \quad -5 \quad -3 \quad -2$$

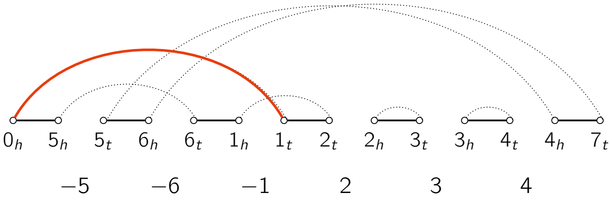
“Unsafe” inversions

- This kind of inversion *always* fixes a breakpoint, increasing the number of cycles by 1.
- But, it is always *good*?
- Not always, because it can create a *bad component*!

Unsafe inversions - Example

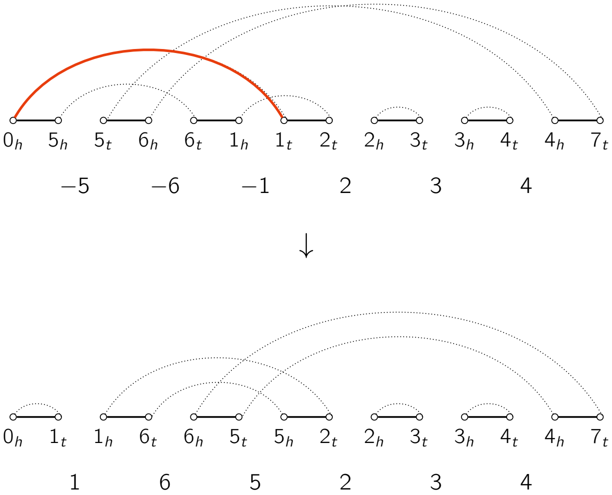


Unsafe inversions - Example



↓

Unsafe inversions - Example



- Increased number of cycles but created a bad component!

How to find safe inversions?

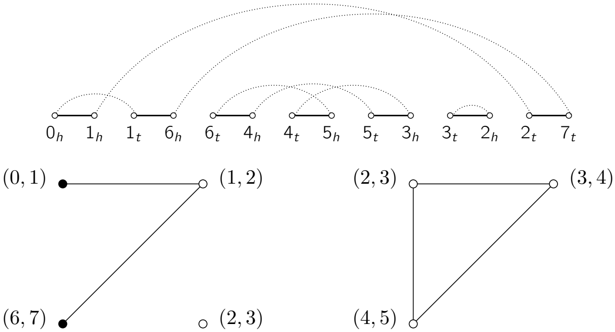
How to know when an inversion is **safe**?
-> Increases the number of cycles *without creating bad components*?

Safe inversions - Definitions

- The **score** of a inversion is the number of *oriented edges* in the BP graph, *after* the application of the reversal.

Safe inversions - Definitions

- The **score** of a inversion is the number of *oriented edges* in the BP graph, *after* the application of the reversal.
- In the last example, the resulting BP and Overlap graphs are:



The score of that reversal is 2.

Safe inversions - Definitions

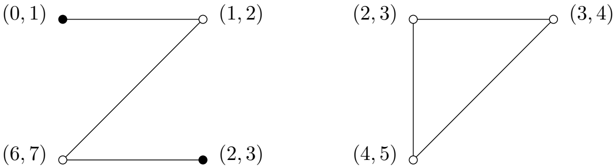
Definition (Inversion score)

The score of a inversion induced by a vertex v in the overlap graph is given by

$$s(v) = T + U - O - 1$$

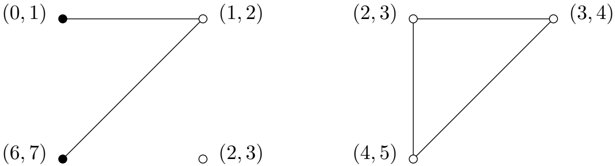
where T is the number of oriented vertices in the graph, U and O are the number of unoriented and oriented vertices adjacent to v, respectively.

Inversion Score - example



For $v = (2, 3)$, we have $T = 2$, $U = 1$, $O = 0$. Therefore $s(v) = T + U - O - 1 = 2$.

After applying the inversion, we have the following graph:



and we see that the score (number of oriented vertices) is indeed 2.

Safe inversions

- **Safe inversions** are inversions that increase the number of cycles of the BP graph by one and do not create new unoriented components.
- Can we always find safe inversions?

Safe inversions

- **Safe inversions** are inversions that increase the number of cycles of the BP graph by one and do not create new unoriented components.
- Can we always find safe inversions? Yes:

Theorem (Bergeron, 2001)

Among all possible oriented inversions, an inversion of maximal score is always safe.

Safe inversions

- **Safe inversions** are inversions that increase the number of cycles of the BP graph by one and do not create new unoriented components.
- Can we always find safe inversions? Yes:

Theorem (Bergeron, 2001)

Among all possible oriented inversions, an inversion of maximal score is always safe.

- **Algorithm**: Apply maximal score inversions until all components are sorted.

Example

$$\pi = (0 \quad 3 \quad 1 \quad 6 \quad 5 \quad -2 \quad 4 \quad 7)$$

For the next lecture

- These slides will be soon available in the Wiki page.

- Don’t forget the exercises!