582670 Algorithms for Bioinformatics

Lecture 5: Combinatorial Algorithms and Genomic Rearrangements

1.10.2015

Adapted from slides by Alexandru Tomescu,
Leena Salmela, Veli Mäkinen, Esa Pitkänen

Background

- We now have genomes of several species available
- It is possible to compare genomes of two or more different species
  $\Longrightarrow$ Comparative genomics
  - See Lecture 4
- Basic observation:
  - Closely related species (such as human and mouse) can be almost identical in terms of genomic content...
  - ... but the order of genomic segments can be very different between species

Example: Human vs. Mouse

- Human chromosome 6 contains elements from six different mouse chromosomes
- In chromosome X the rearrangements are mostly within the same chromosome

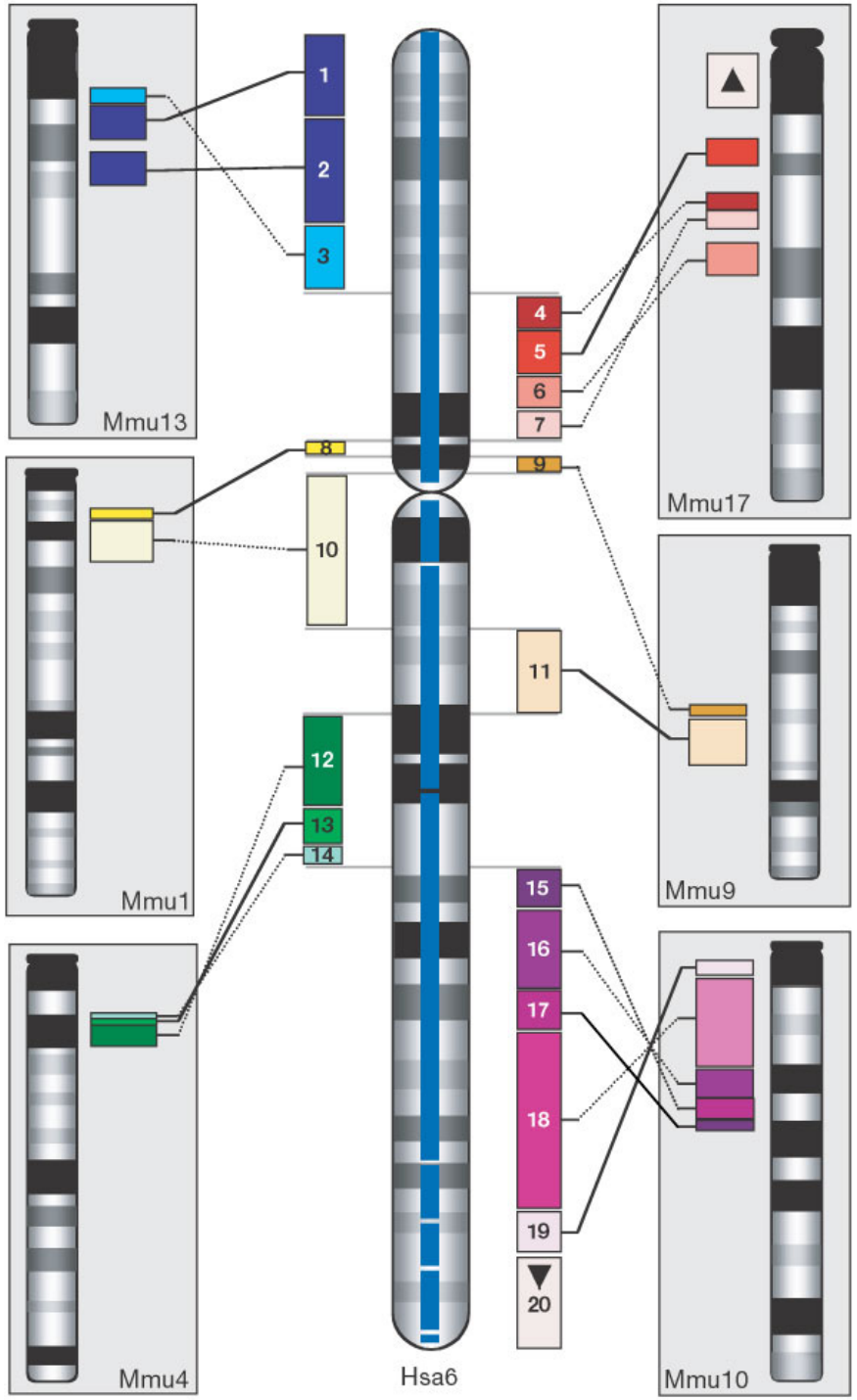


Image from: Gregory et al.: A physical map of the mouse genome. Nature 418, 743-750 (15 August 2002).

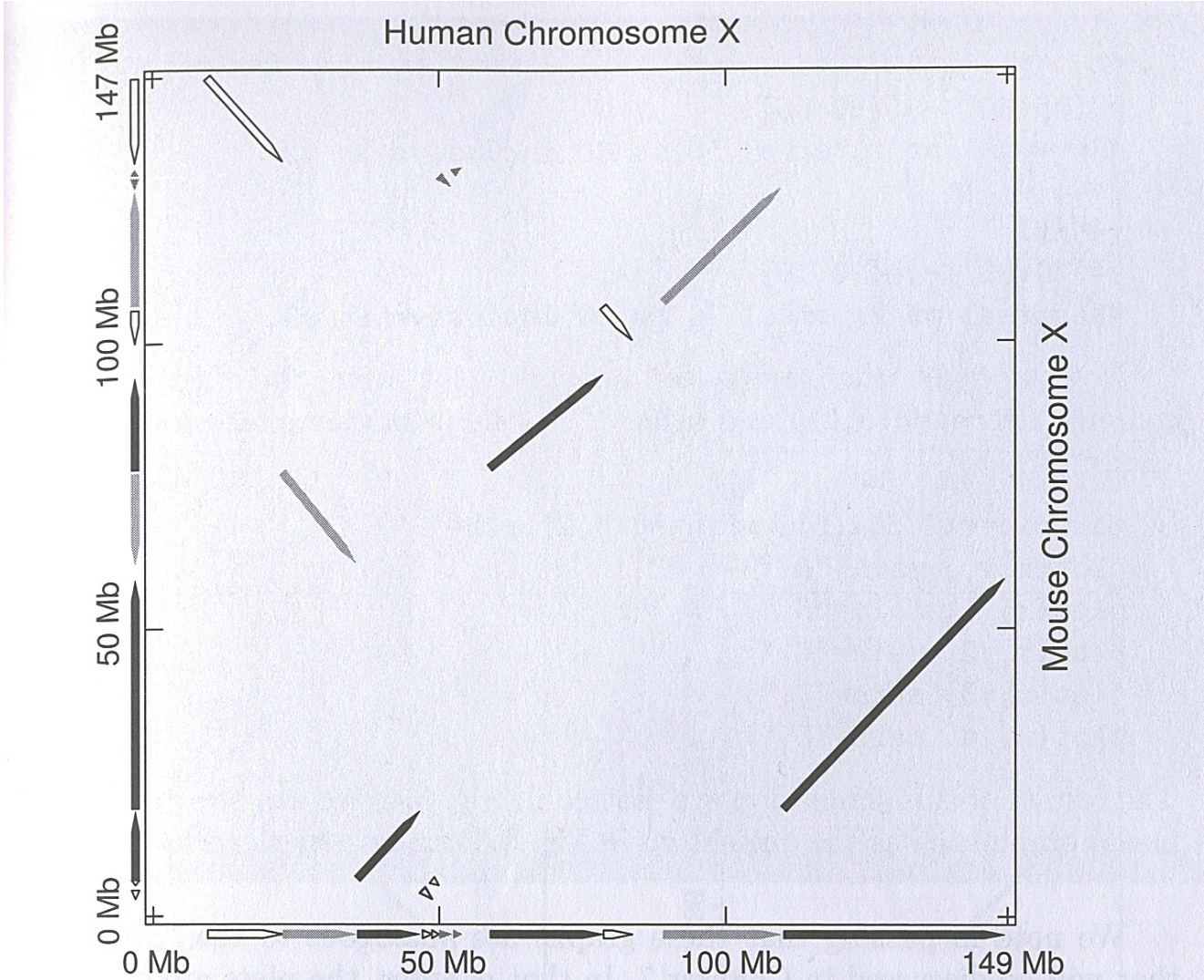


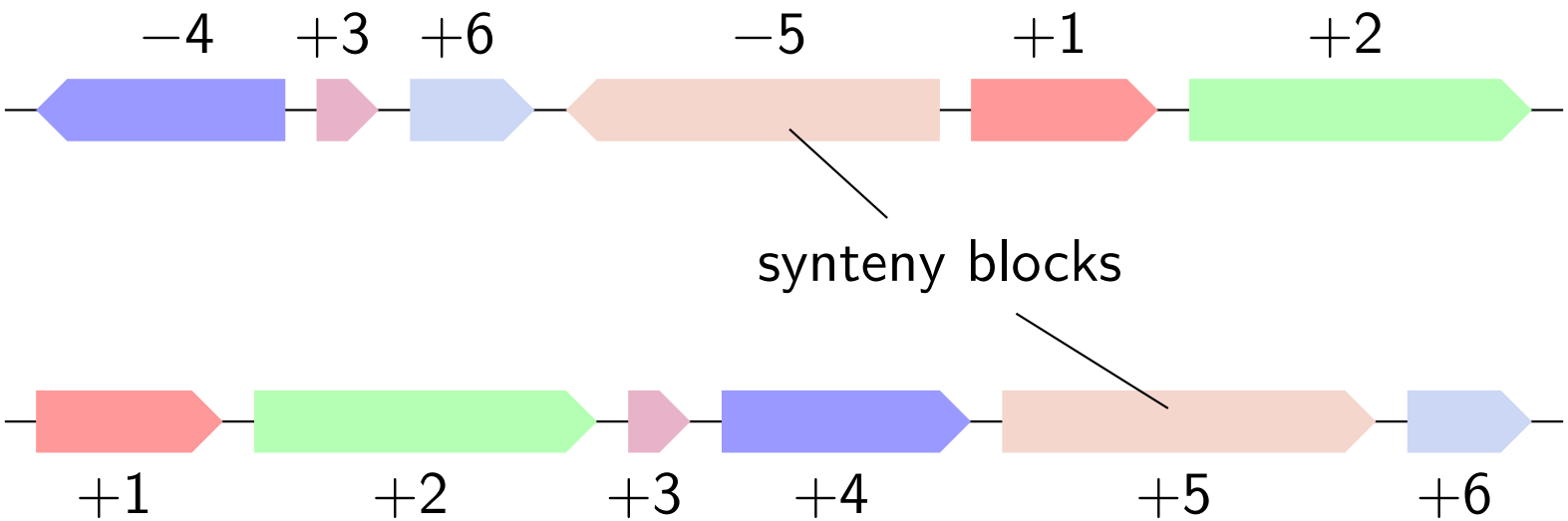
Fig. 5.3. Synteny blocks shared by human and mouse X chromosomes. The arrowhead for each block indicates the direction of increasing coordinate values for the human X chromosome. Reprinted, with permission, from Pevzner P and Tesler G (2003) *Genome Research* 13:37–45. Copyright 2003 Cold Spring Harbor Laboratory Press.

Genomic Rearrangements

- The differences result from genomic rearrangement events
  - Rare evolutionary events
- The number of such events can be used for estimating the evolutionary distance between species
- Problem: What is the minimum number of events needed to rearrange one genome into the other?
  - A kind of edit distance

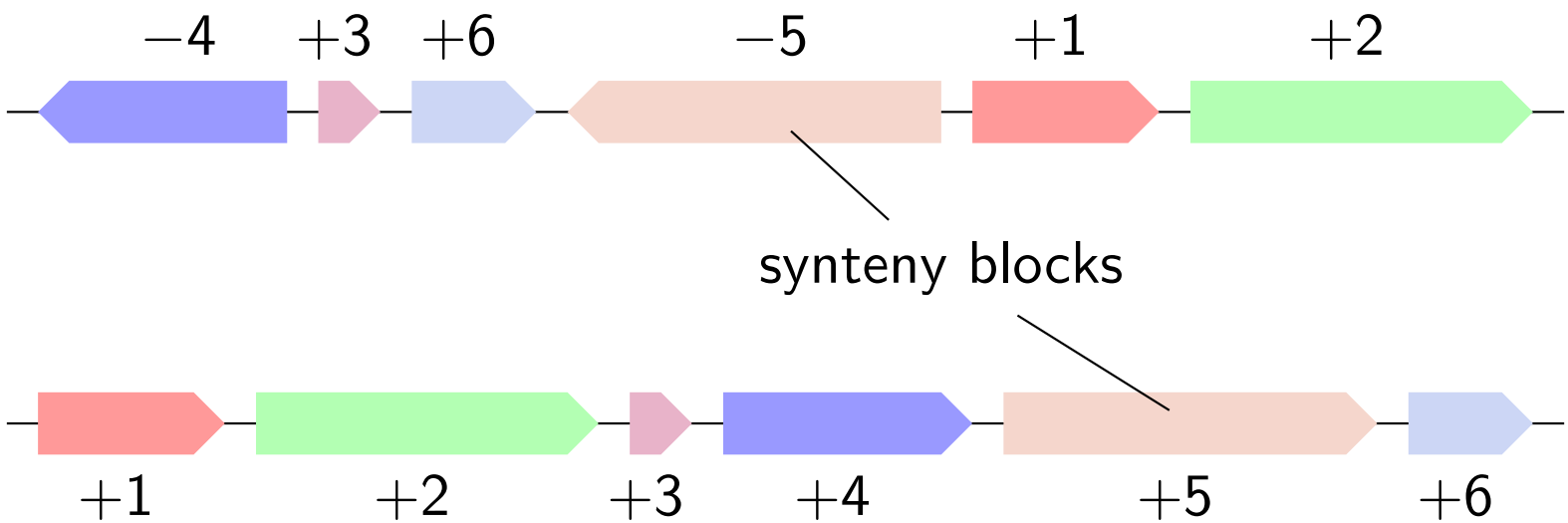
Synteny Blocks

- Synteny block = similar region in two different genomes
  - Contain homologous sequences and usually genes with similar function
  - Different locations, possibly even different chromosomes
  - Possibly different orientations (i.e., different strands)



Signed Permutations

- Assign numbers $1, 2, \ldots, n$ to the synteny blocks
- Assign signs ($+$ or $-$) to denote orientation
- A genome is then represented by a sequence of n signed numbers called a *signed permutation*
  - Permutation because each (unsigned) number appears exactly once



Sorting permutations

- The problem is then to convert one signed permutation to the other
  - Using what operations/events? (next slide)
- By convention, the numbering and signs are chosen so that one of the genomes is represented by the *identity permutation* $(+1 + 2 \cdots + n)$
- Then we want to convert the other signed permutation into the identity permutation
  - This is called *sorting* the permutation

Reversals

- The most common genomic rearrangement event is reversal
- A contiguous section of a chromosome is reversed
- The orientation (sign) of all synteny blocks within the section changes

$$\begin{array}{cccccc} -4 & +3 & +6 & -5 & +1 & +2 \\ \hline -4 & +5 & -6 & -3 & +1 & +2 \end{array}$$

Sorting by Reversals problem

- Goal: Find the shortest *series of reversals* that tranforms a given signed permutation to the *identity* permutation
- Input: Signed permutation P of the numbers $1, \ldots, n$
- Output: A series of reversals that transforms P into $(+1\ +2\ \cdots\ +n)$.
- Objective: Minimize the number of reversals
- The smallest possible number of reversals is called the *reversal distance* and is denoted by $d_{\text{rev}}(P)$.

Sorting by Reversals: Example

$$
\begin{array}{cccccc}
-4 & +3 & +6 & -5 & +1 & +2 \\
\hline
-4 & +3 & -2 & -1 & +5 & -6 \\
\hline
+1 & +2 & -3 & +4 & +5 & -6 \\
\hline
+1 & +2 & +3 & +4 & +5 & -6 \\
\hline
+1 & +2 & +3 & +4 & +5 & +6
\end{array}
$$

- This shows that $d_{\text{rev}}(-4\ +3\ +6\ -5\ +1\ +2) \leq 4$
- Is four the smallest number of reversals?

Greedy reversal sort

- Greedy reversal sort algorithm
  - Move 1 to correct location and orientation
  - Move 2 to correct location and orientation without moving 1 again
  - Move 3 to correct location and orientation without moving 1 or 2
  - etc.
- Resembles selection sort

Greedy reversal sort: Example

$$
\begin{array}{cccccc}
-4 & +3 & +6 & -5 & +1 & +2 \\
\hline
-1 & +5 & -6 & -3 & +4 & -6 \\
\hline
+1 & +5 & -6 & -3 & +4 & +2 \\
\hline
+1 & -2 & -4 & +3 & +6 & -5 \\
\hline
+1 & +2 & -4 & +3 & +6 & -5 \\
\hline
+1 & +2 & -3 & +4 & +6 & -5 \\
\hline
+1 & +2 & +3 & +4 & +6 & -5 \\
\hline
+1 & +2 & +3 & +4 & +5 & -6 \\
\hline
+1 & +2 & +3 & +4 & +5 & +6
\end{array}
$$

How good is greedy reversal sort?

- Not so good
- In our example, greedy reversal sort needed 8 reversals but we know that $d_{\text{rev}} \leq 4$
- Even worse case is $P = (-6\ +1\ +2\ +3\ +4\ +5)$
  - Greedy reversal sort needs 10 reversals but $d_{\text{rev}}(P) = 2$

-6	$+1$	$+2$	$+3$	$+4$	$+5$
-5	-4	-3	-2	-1	$+6$
$+1$	$+2$	$+3$	$+4$	$+5$	$+6$

- For any n, there is a permutation P of length n for which greedy reversal sort requires at least $(n-1) \cdot d_{\text{rev}}(P)$ reversals

Approximation algorithms and approximation ratios

- Greedy reversal sort is an *approximation algorithm*. It only produces an approximate solution.
- $\mathcal{A}(P)$: approximate solution returned by algorithm $\mathcal{A}$
- $OPT(P)$: optimal solution
- The *approximation ratio* of (minimization) algorithm $\mathcal{A}$ is the maximum approximation ratio over *all* inputs of size n:

$$\max_{|P|=n} \frac{\mathcal{A}(P)}{OPT(P)}$$

- The approximation ratio for greedy reversal sort is thus at least $(n-1)$

Breakpoints and Adjacencies

- Consider a permutation $P = (p_1\ p_2\ \cdots\ p_n)$
- Add $p_0 = 0$ to the beginning and $p_{n+1} = +(n+1)$ to the end
- Each consecutive pair $(p_i\ p_{i+1})$ is
  - *Adjacency* if $p_{i+1} - p_i = 1$
  - *Breakpoint* otherwise

- Example $P = (+3\ +4\ +5\ -1\ -2\ -7\ -6\ +8)$

$$0 \mid +3\ \ +4\ \ +5 \mid -1 \mid -2 \mid -7\ \ -6 \mid +8\ \ +9$$

Adjacencies: $(+3\ +4)$, $(+4\ +5)$, $(-7\ -6)$, $(+8\ +9)$
Breakpoints: $(0\ +3)$, $(+5\ -1)$, $(-1\ -2)$, $(-2\ -7)$, $(-6\ +8)$

- In the identity permutation $(+1 \cdots +n)$, all pairs are adjacencies

Reversals and breakpoints

How a reversal affects a pair $(p_i\ p_{i+1})$

- No change if $(p_i\ p_{i+1})$ is outside the reversal region
- Reversal and inversion of signs if $(p_i\ p_{i+1})$ is inside the region
  - $(p_i\ p_{i+1}) \Rightarrow (-p_{i+1}\ -p_i)$
  - But no change to breakpoint status since $p_{i+1} - p_i = (-p_i) - (-p_{i+1})$
- The pair is separated if it is on the reversal region boundary

- Thus a reversal changes the breakpoint status of at most two pairs

Breakpoint Theorem

- $\text{Breakpoints}(P) =$ the number of breakpoints in P
- Theorem: For any signed permutation P,

$$d_{\text{rev}}(P) \geq \text{Breakpoints}(P)/2$$

- Proof
  - $\text{Breakpoints}(I) = 0$ for identity permutation I
  - A reversal eliminates at most two breakpoints
  - At least $\text{Breakpoints}(P)/2$ reversals are needed to reduce $\text{Breakpoints}(P)$ to zero

Breakpoint Theorem: Example

- Earlier we saw that $d_{\text{rev}}(P) \leq 4$ for $P = (-4\ +3\ +6\ -5\ +1\ +2)$
- Since $\text{Breakpoints}(P) = 6$, the breakpoint theorem shows that $d_{\text{rev}}(P) \geq 3$
- If $d_{\text{rev}}(P) = 3$, there would have to be a series of three reversals each of which eliminates exactly two breakpoints
  - Is there one?
- Otherwise $d_{\text{rev}}(P) = 4$

How good is Breakpoint Theorem?

- Consider a permutation $P = (+n\ +(n-1)\ \cdots\ +1)$
- Requires $n+1$ reversals to sort: $d_{\text{rev}}(P) = n+1$
- $\text{Breakpoints}(P) = n+1$
- Breakpoint Theorem: $d_{\text{rev}}(P) \geq (n+1)/2$
- Factor of two too small lower bound
  - Is this good?

Breakpoint Reversal Sort

1: **while** $\mathrm{Breakpoints}(P) > 0$ **do**
2: **if** There is a reversal r_2 that reduces $\mathrm{Breakpoints}(P)$ by two **then**
3: Perform r_2
4: **else if** There is a reversal r_1 that reduces $\mathrm{Breakpoints}(P)$ by one **then**
5: Perform r_1
6: **else**
7: Perform a reversal that does not increase $\mathrm{Breakpoints}(P)$

Breakpoint Reversal Sort: Analysis

- If $\mathrm{Breakpoints}(P) > 0$, there is always a reversal that does not increase $\mathrm{Breakpoints}(P)$
  - Reversal boundaries at breakpoints
- If P contains negative signs, there is always a reversal that decreases $\mathrm{Breakpoints}(P)$
  - Proof as exercise
- Thus two consecutive reversals by the algorithm always decreases $\mathrm{Breakpoints}(P)$ by at least one
- The number of reversals is at most $2 \cdot \mathrm{Breakpoints}(P)$
- Since $d_{\mathrm{rev}}(P) \geq \mathrm{Breakpoints}(P)/2$, the approximation ratio is at most 4
  - Is this good?

Computing reversal distance

- The breakpoint analysis gives lower and upper bounds for reversal distance that are no more than a factor of four apart
  - Often much closer bounds
- There exists a linear time algorithm for computing the reversal distance but it is much more complicated (Bader, Moret & Yan, 2001)
- However, allowing operations other than reversals makes the problem easier!

Other genomic rearrangements

- Fission: Split a chromosome into two
- Fusion: Concatenate two chromosomes into one
- Translocation: Split two chromosomes and join them together differently

$$(+1\ +2\ +3\ +4\ +5)\ (+6\ +7\ +8\ +9)$$

$$\Downarrow$$

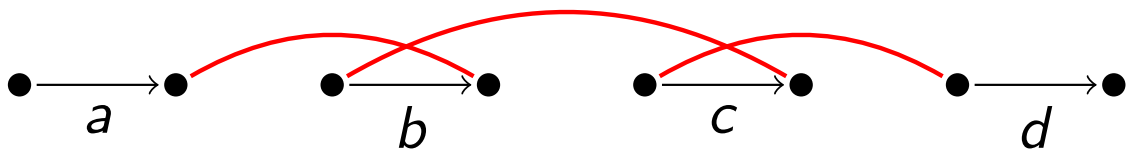
$$(+1\ +2\ +8\ +9)\ (+6\ +7\ +3\ +4\ +5)$$

  - More common than fusion or fission

Synteny Block Graph

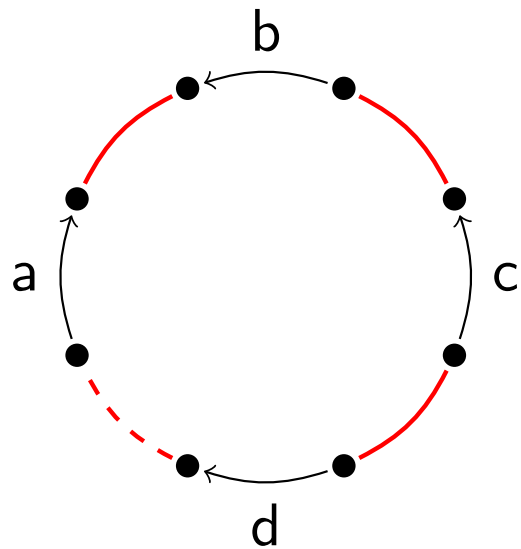
- By convention, the synteny blocks are now labelled with letters rather than numbers
  - $+$ and $-$ signs are still used
- For each synteny block, the graph has two nodes connected by a directed edge indicating the orientation of the block
- For each chromosome, the directed edges are connected by undirected edges in the order they appear in the chromosome
- Example: $P = (+a\ -b\ -c\ +d)$

- Nodes can be moved around without losing information



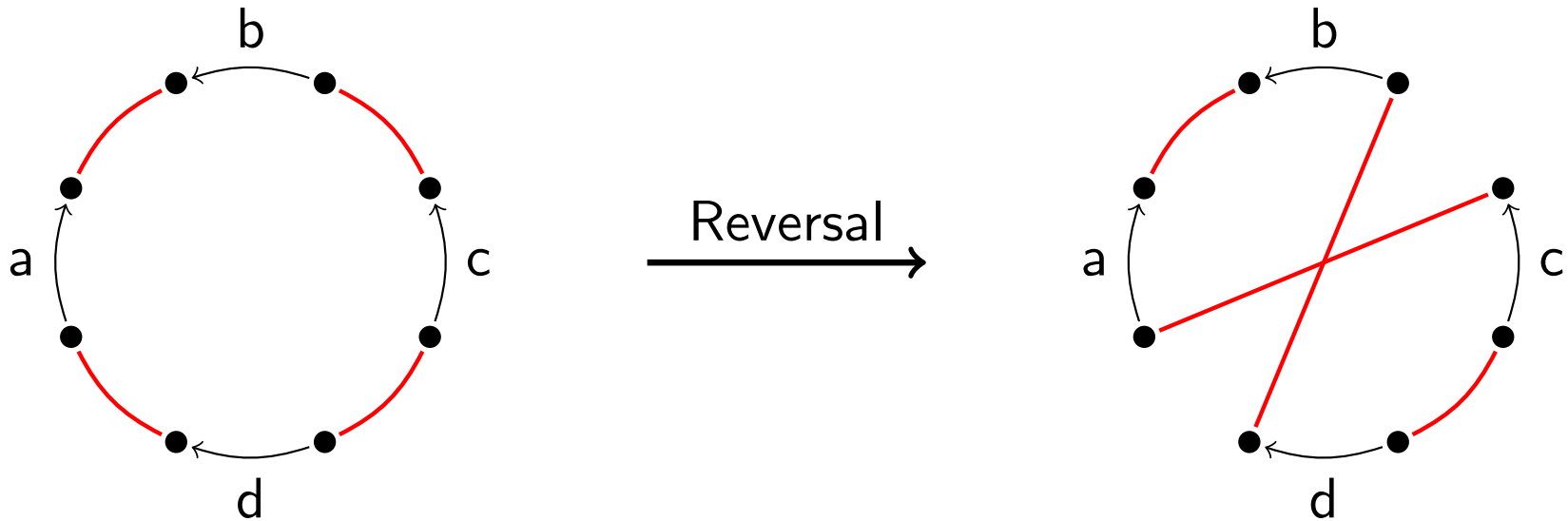
Cyclic Synteny Block Graph

- Arrange block edges on a circle
- Add a connecting edge to complete the cycle
  - As if the chromosome was circular
- Example: $P = (+a\ -b\ -c\ +d)$



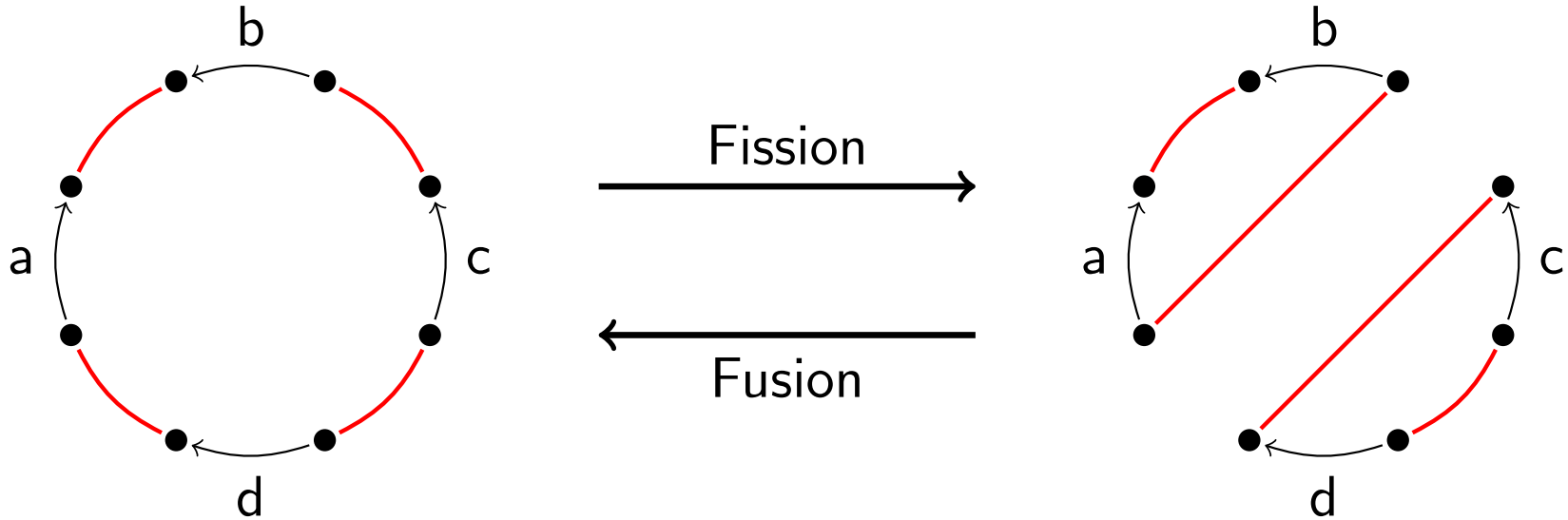
Reversal in Synteny Block Graph

- Reversal operation replaces a pair of connecting edges with a different pair of edges between the same nodes
- Example: Reversal transform of $(+a\ -b\ \underline{-c\ +d})$ into $(+a\ -b\ -d\ +c)$



Fission and Fusion in Synteny Block Graph

- Fission and fusion too are similar edge replacements
- Example: Fission of $(+a\ -b\ -c\ +d)$ into $(+a\ -b)$ $(-c\ +d)$ and the inverse fusion

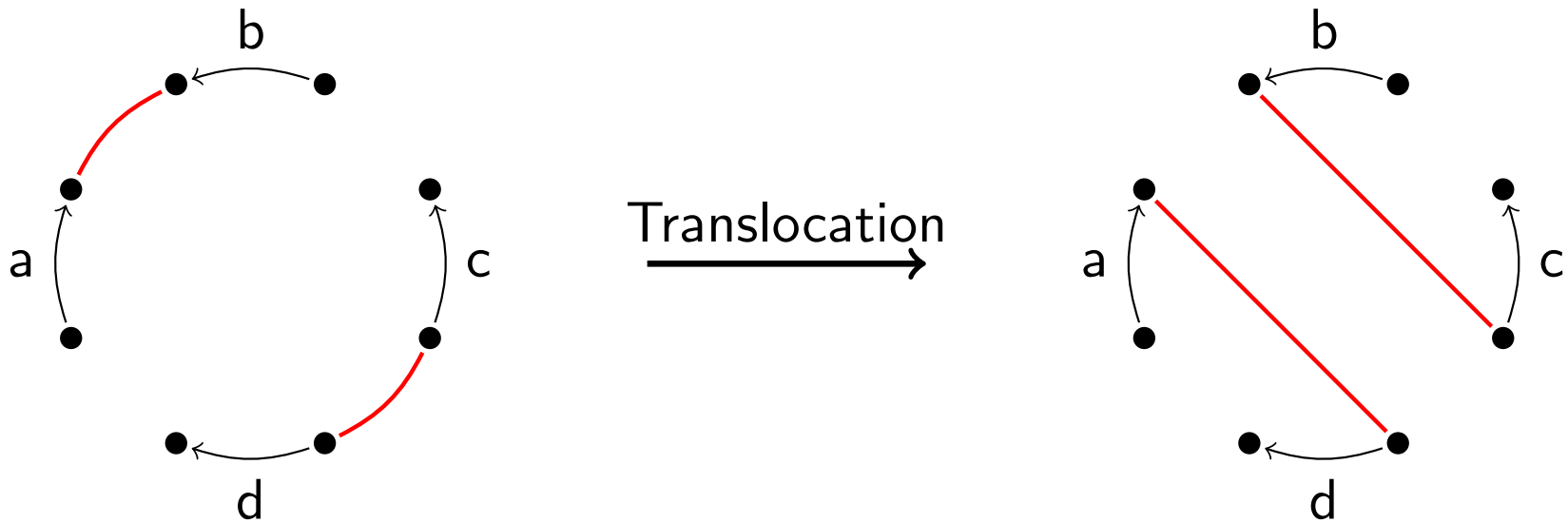


2-Breaks

- The basic synteny block graph operation is 2-break:
  - Remove two connecting (undirected) edges
  - Add two edges connecting the same four nodes differently
- As we saw, a single 2-break can cause a reversal, a fission or a fusion

Translocation in Synteny Block Graph

- Translocation too can be implemented by a 2-break but on a graph without the edges that connect chromosome endpoints
- Example: Translocation of $(+a\ -b)\ (-c\ +d)$ into $(+a\ +d)\ (-c\ -b)$



2-Break Distance

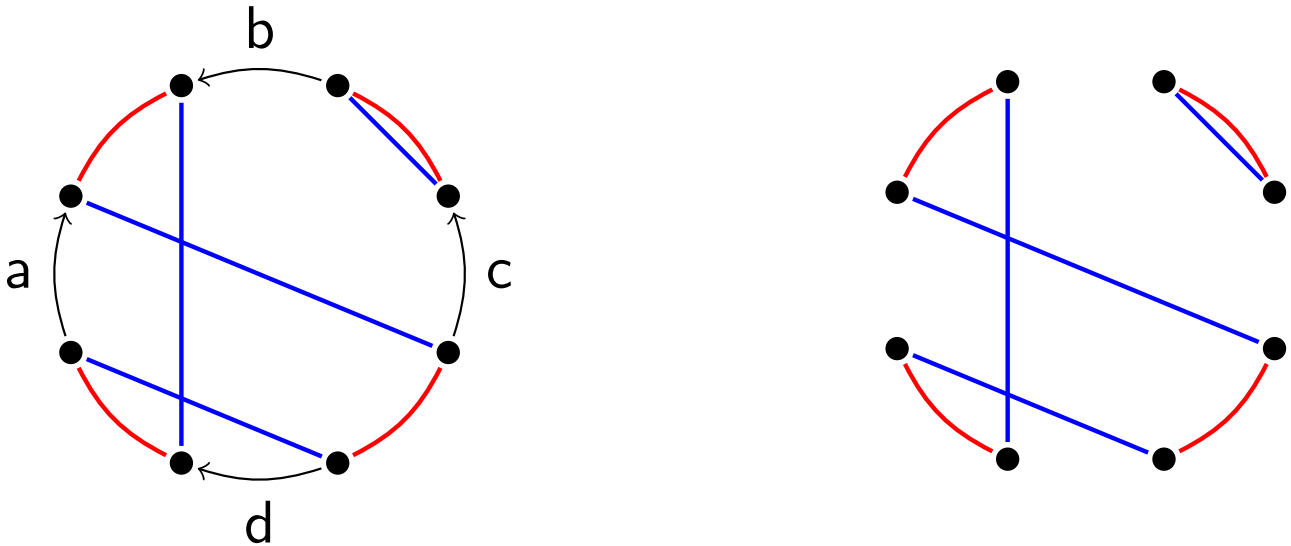
- For simplicity, we consider only fully cyclic graphs
  - Corresponds to genomes with circular chromosomes
- We are interested in the minimal number of 2-breaks needed to transform one graph into the other
  - Corresponds to the minimal number of reversals, fissions and fusions to transform one genome to the other
  - This number is the *2-break distance*

2-Break Distance Problem

- Goal: Compute the 2-break distance between genomes
- Input: Two genomes P and Q with circular chromosomes on the same set of synteny blocks
- Output: The 2-break distance $d(P, Q)$ between P and Q

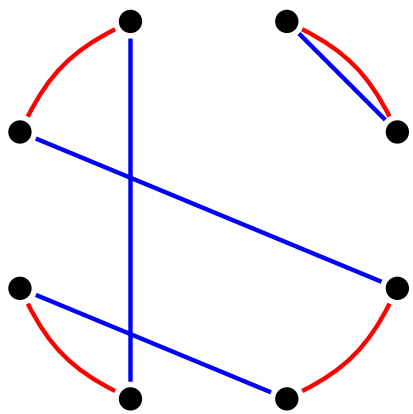
Breakpoint Graph

- The breakpoint graph $\mathrm{Breakpoint}(P, Q)$ is a merging of the synteny block graphs for P and Q
  - Shared block edges (which can be omitted after construction)
  - Separate connecting edges
- Example: $\mathrm{Breakpoint}(P, Q)$ for $P = (+a\ -b\ -c\ +d)$ and $Q = (+a\ +c\ +b\ -d)$



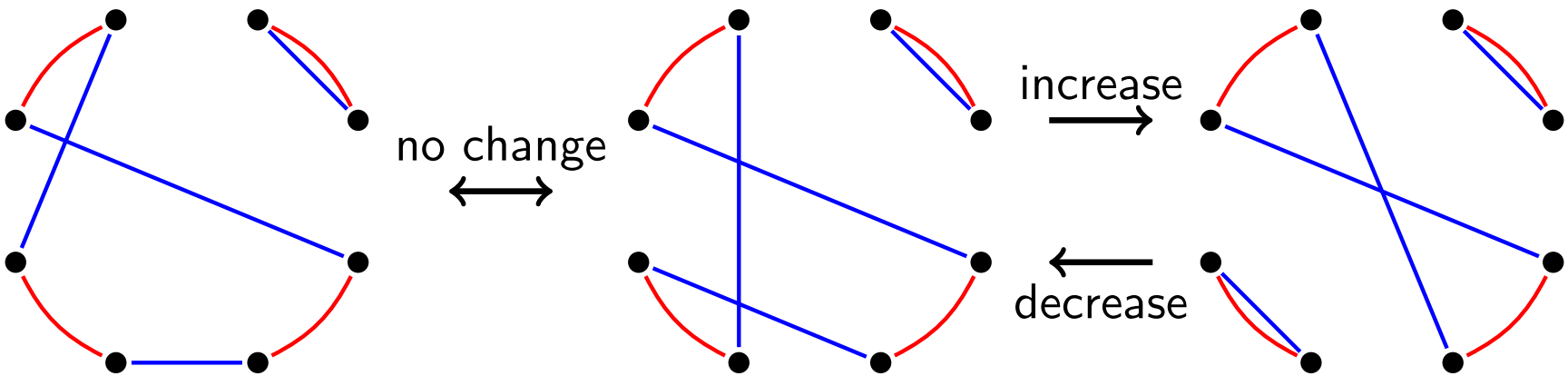
Cycles in Breakpoint Graph

- A breakpoint graph contains cycles with alternating red and blue edges
- Let $\mathrm{Cycles}(P, Q)$ denote the number of such cycles in $\mathrm{Breakpoint}(P, Q)$
- Example: $\mathrm{Cycles}(P, Q) = 2$ for $P = (+a\ -b\ -c\ +d)$ and $Q = (+a\ +c\ +b\ -d)$



Cycles and 2-Breaks

- A 2-break can increase the number of cycles by one, decrease the number of cycles by one or not change the number of cycles



- Note that the alternating red–blue cycles are different from the cycles representing chromosomes.
- Thus a 2-break splitting a red-blue cycle does not necessarily represent a fission.

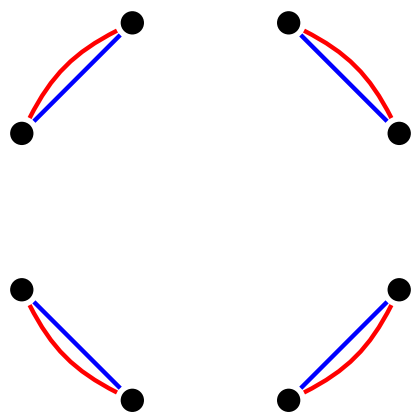
Cycle Theorem

- Theorem: For any genomes P and Q, any 2-break applied to P or to Q can increase $\mathrm{Cycles}(P, Q)$ by at most one.

- Proof
  - A 2-break removes two edges and adds two edges.
  - The first removal of an edge breaks a cycle.
    (The second removal might be in the same cycle.)
  - Each addition of an edge can create at most one cycle.
  - The total increase in the number of cycles is at most $-1 + 2 = 1$

Cycles in Identical Genomes

- Let $\mathrm{Blocks}(P, Q)$ be the number of synteny blocks (shared by P and Q)
- $\mathrm{Cycles}(P, P) = \mathrm{Blocks}(P, Q)$



Splitting Cycles

- A cycle longer than two can always be split into two cycles by a 2-break.
  - Remove any two edges on the cycle and replace them appropriately.
- If $\mathrm{Cycles}(P, Q) < \mathrm{Blocks}(P, Q)$, there must be at least one cycle longer than two, and thus there exists a 2-break that increases the number of cycles.

Cycles and 2-Break Distance

- Theorem: For any genomes P and Q, $d(P, Q) = \text{Blocks}(P, Q) - \text{Cycles}(P, Q)$.

- Proof
  - Consider a series of 2-breaks that change Q into P
  - The series must have at least $\text{Cycles}(P, P) - \text{Cycles}(P, Q)$ 2-breaks, because each 2-break can add at most one cycle.
  - On the other hand, there exists a series of no more than $\text{Cycles}(P, P) - \text{Cycles}(P, Q)$ 2-breaks, because we can always find a 2-break that increases the number of cycles.
  - Thus $d(P, Q) = \text{Cycles}(P, P) - \text{Cycles}(P, Q) = \text{Blocks}(P, Q) - \text{Cycles}(P, Q)$.

2-Break Distance vs. Reversal Distance

- 2-break distance is simple to compute
- Reversal distance is complicated to compute
  - Study groups
- Adding the extra operations, fission and fusion, helped make the problem simpler