CS481: Bioinformatics Algorithms

Can Alkan
EA509
calkan@cs.bilkent.edu.tr

http://www.cs.bilkent.edu.tr/~calkan/teaching/cs481/

GENOME REARRANGEMENTS

Turnip vs Cabbage: Look and Taste Different

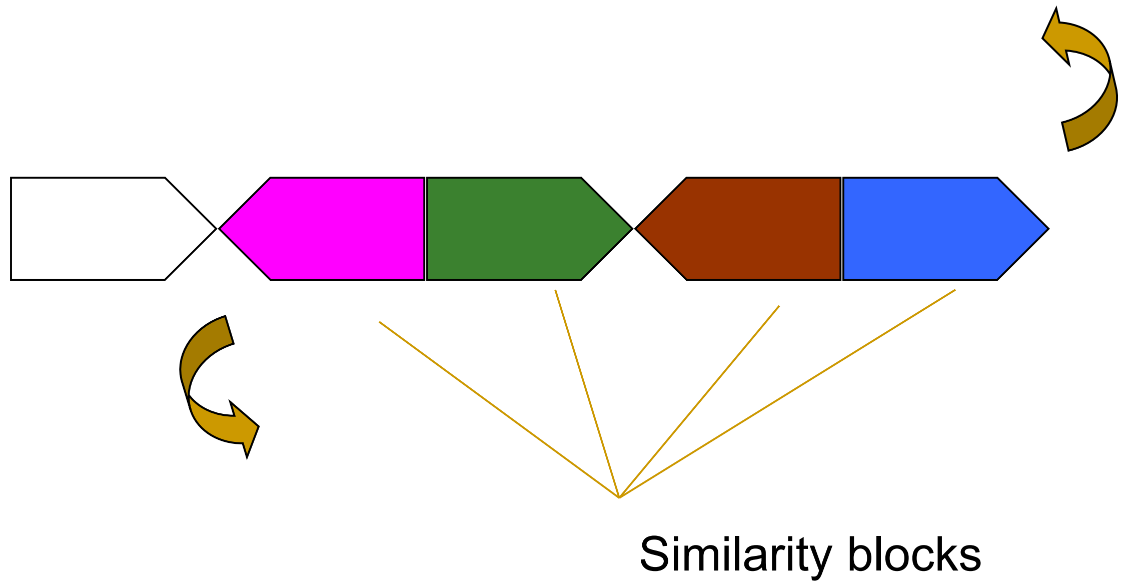
- Although cabbages and turnips share a recent common ancestor, they look and taste different

Turnip vs Cabbage: Almost Identical mtDNA gene sequences

- In 1980s Jeffrey Palmer studied evolution of plant organelles by comparing mitochondrial genomes of the cabbage and turnip
- 99% similarity between genes
- These surprisingly identical gene sequences differed in gene order
- This study helped pave the way to analyzing genome rearrangements in molecular evolution

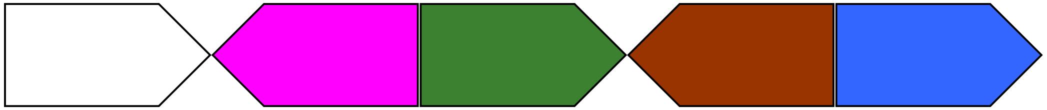
Turnip vs Cabbage: Different mtDNA Gene Order

- Gene order comparison:



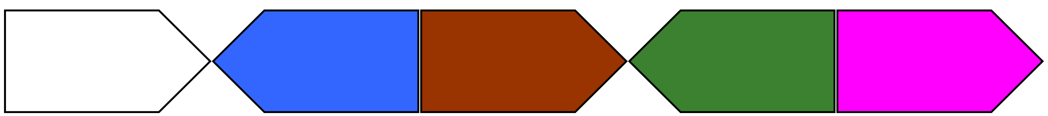
Turnip vs Cabbage: Different mtDNA Gene Order

- Gene order comparison:



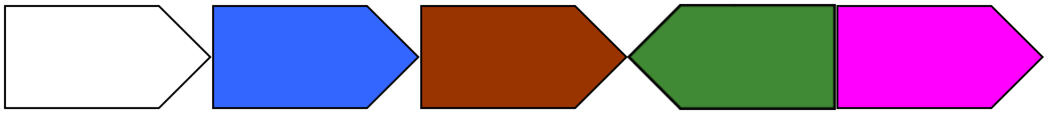
Turnip vs Cabbage: Different mtDNA Gene Order

- Gene order comparison:



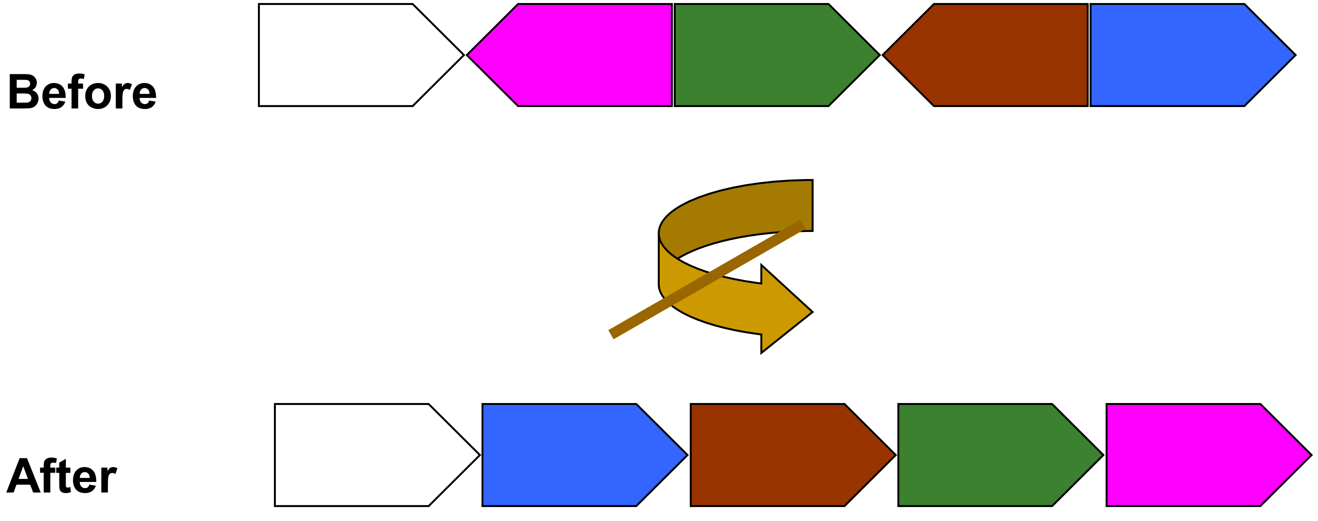
Turnip vs Cabbage: Different mtDNA Gene Order

- Gene order comparison:



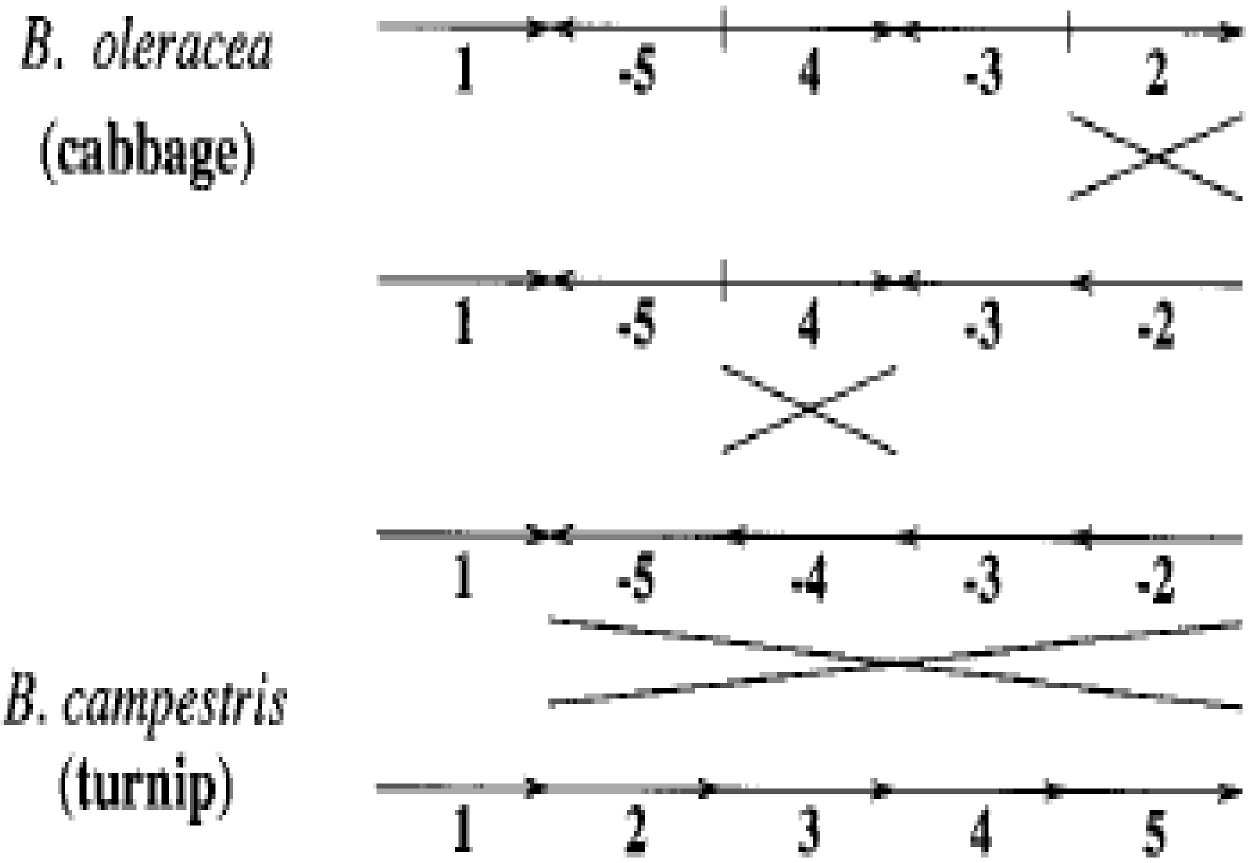
Turnip vs Cabbage: Different mtDNA Gene Order

- Gene order comparison:

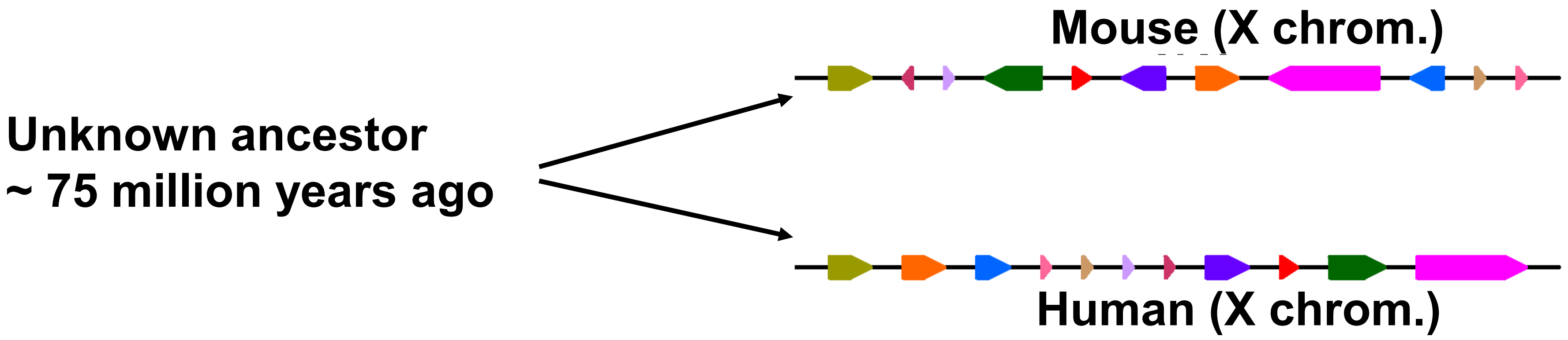


Evolution is manifested as the divergence in gene order

Transforming Cabbage into Turnip

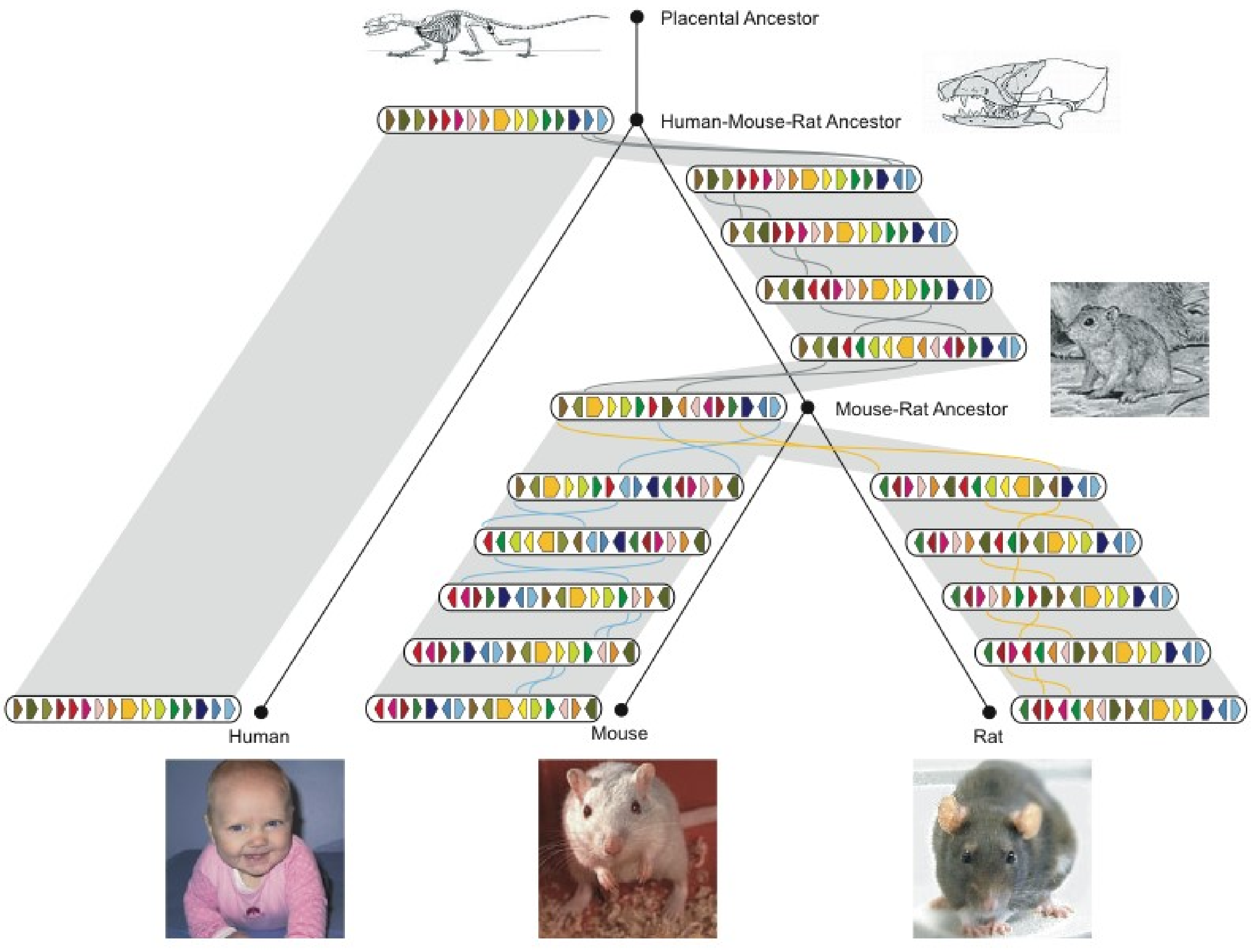


Genome rearrangements



- What are the similarity blocks and how to find them?
- What is the architecture of the ancestral genome?
- What is the evolutionary scenario for transforming one genome into the other?

History of Chromosome X



Rat Consortium, *Nature*, 2004

articles

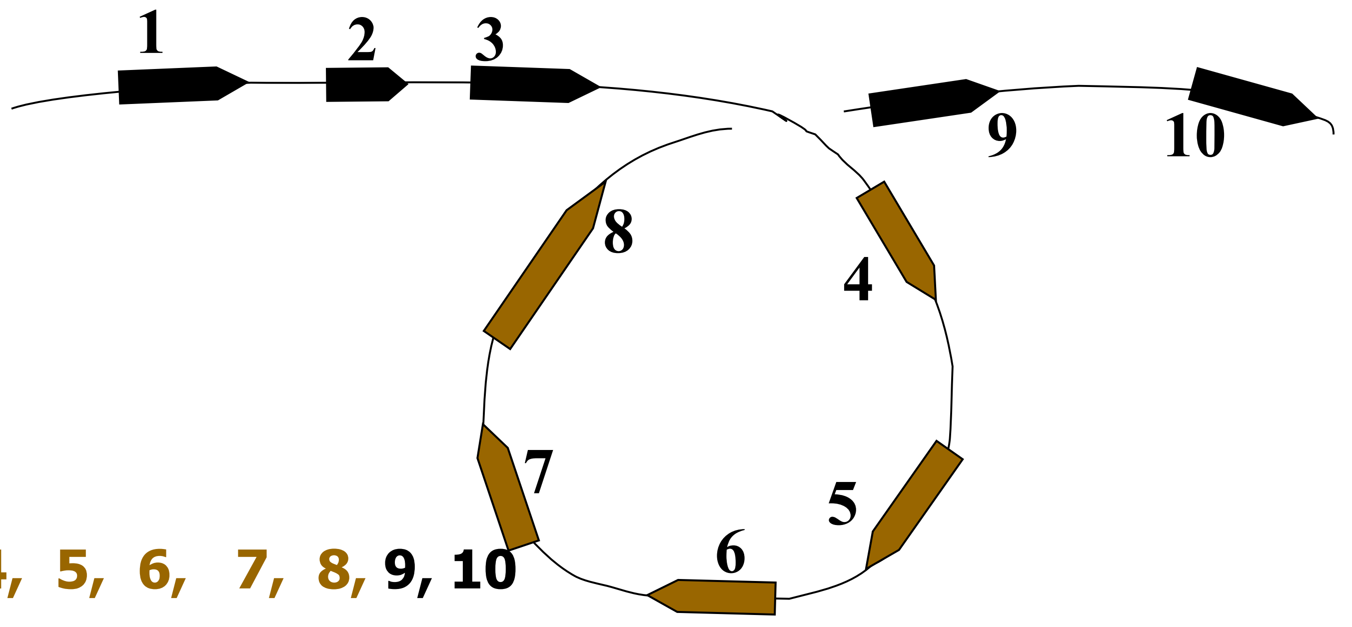
Genome sequence of the Brown Norway rat yields insights into mammalian evolution

Rat Genome Sequencing Project Consortium*

**Lists of participants and affiliations appear at the end of the paper*

Analysis and annotation: Affymetrix Simon Cawley[19]; **Baylor College of Medicine** George M. Weinstock (Coordinator)[1], Kim C. Worley (Overall Coordinator)[1], A. J. Cooney[20], Richard A. Gibbs[1], Lisa M. D'Souza[1], Kirt Martin[1], Jia Qian Wu[1], Manuel L. Gonzalez-Garay[1], Andrew R. Jackson[1], Kenneth J. Kalafus[1,58], Michael P. McLeod[1], Aleksandar Milosavljevic[1], Davinder Virk[1], Andrei Volkov[1], David A. Wheeler[1], Zhengdong Zhang[1]; **Case Western Reserve University** Jeffrey A. Bailey[4], Evan E. Eichler[4], Eray Tuzun[4]; **EBI, Wellcome Trust Genome Campus** Ewan Birney[21], Emmanuel Mongin[21], Abel Ureta-Vidal[21], Cara Woodwark[21]; **EMBL, Heidelberg** Evgeny Zdobnov[22], Peer Bork[22,23], Mikita Suyama[22], David Torrents[22]; **Fraunhofer-Chalmers Research Centre for Industrial Mathematics, Gothenburg** Marina Alexandersson[24];

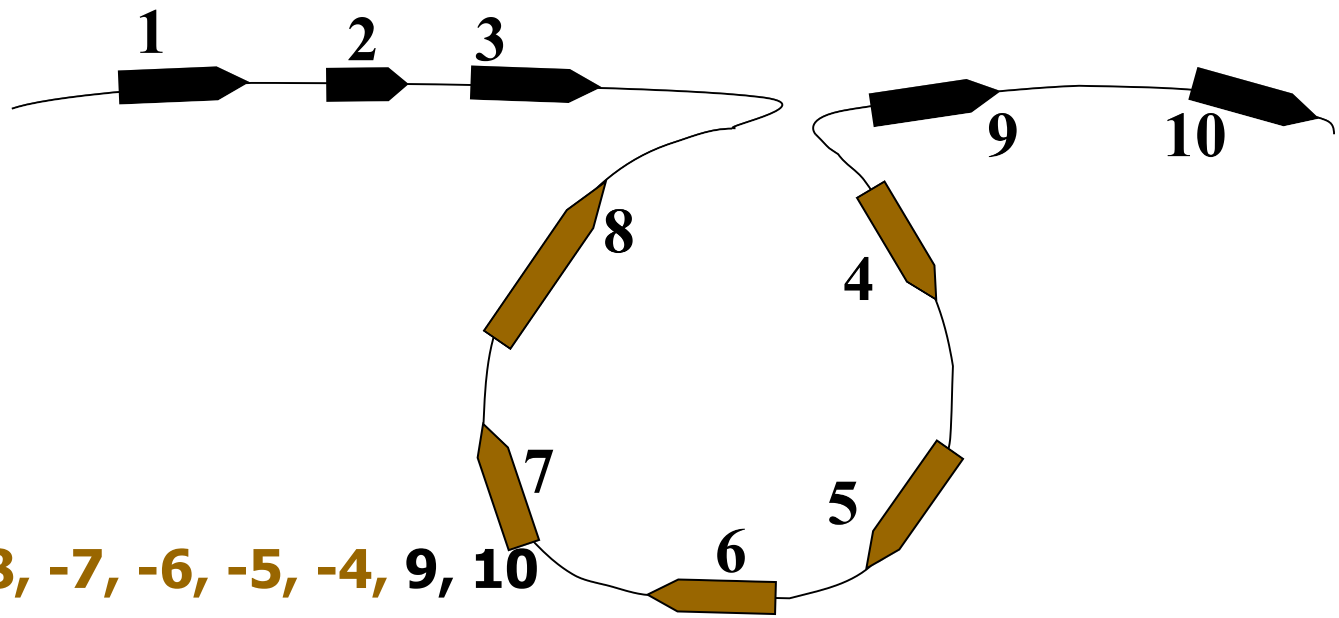
Reversals



1, 2, 3, 4, 5, 6, 7, 8, 9, 10

- Blocks represent conserved genes.

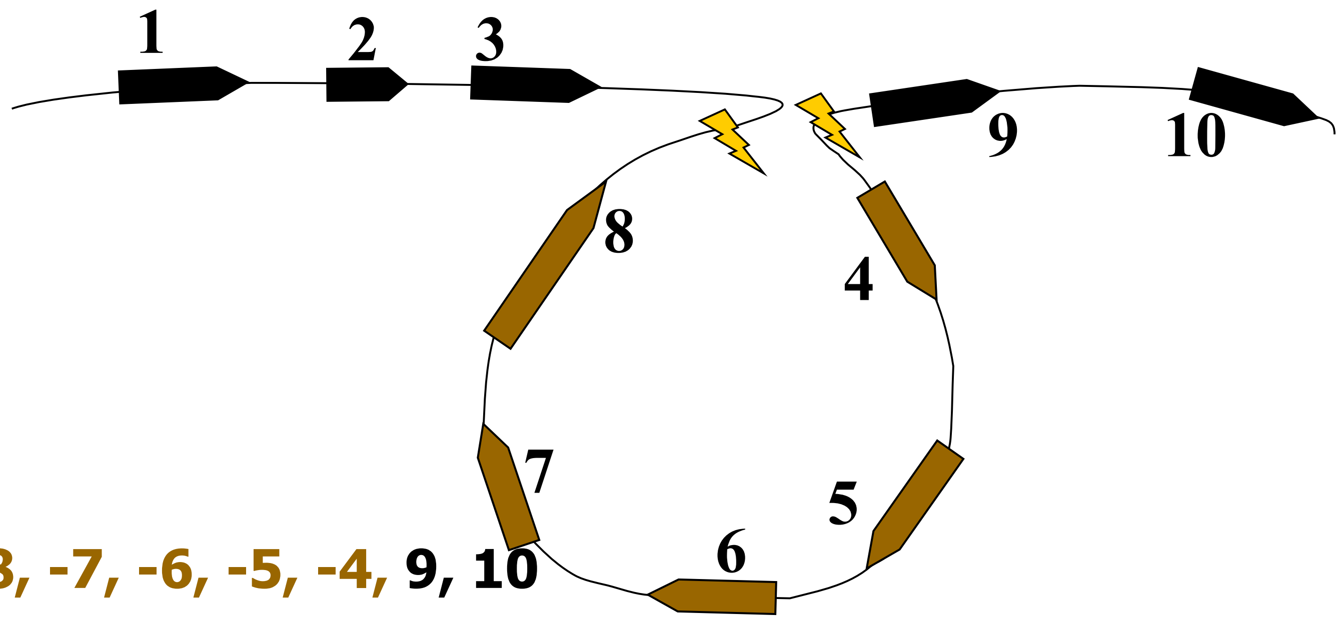
Reversals



1, 2, 3, -8, -7, -6, -5, -4, 9, 10

- **Blocks represent conserved genes.**
- **In the course of evolution or in a clinical context, blocks 1,…,10 could be misread as 1, 2, 3, -8, -7, -6, -5, -4, 9, 10.**

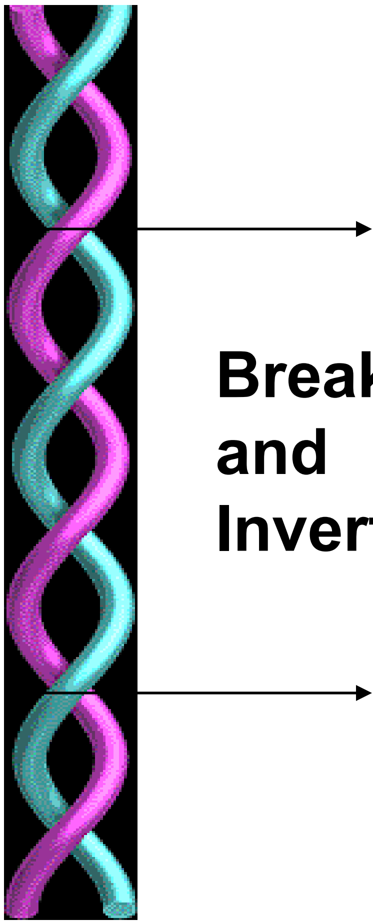
Reversals and Breakpoints



1, 2, 3, -8, -7, -6, -5, -4, 9, 10

The reversion introduced two *breakpoints* (disruptions in order).

Reversals: Example



Break and Invert

5' ATGCCTGTACTA 3'

3' TACGGACATGAT 5'

5' ATGTACAGGCTA 3'

3' TACATGTCCGAT 5'

Types of Rearrangements

Reversal

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

Translocation

1 2 3 $\rightarrow$ 1 2 6

4 5 6 $\rightarrow$ 4 5 3

Fusion

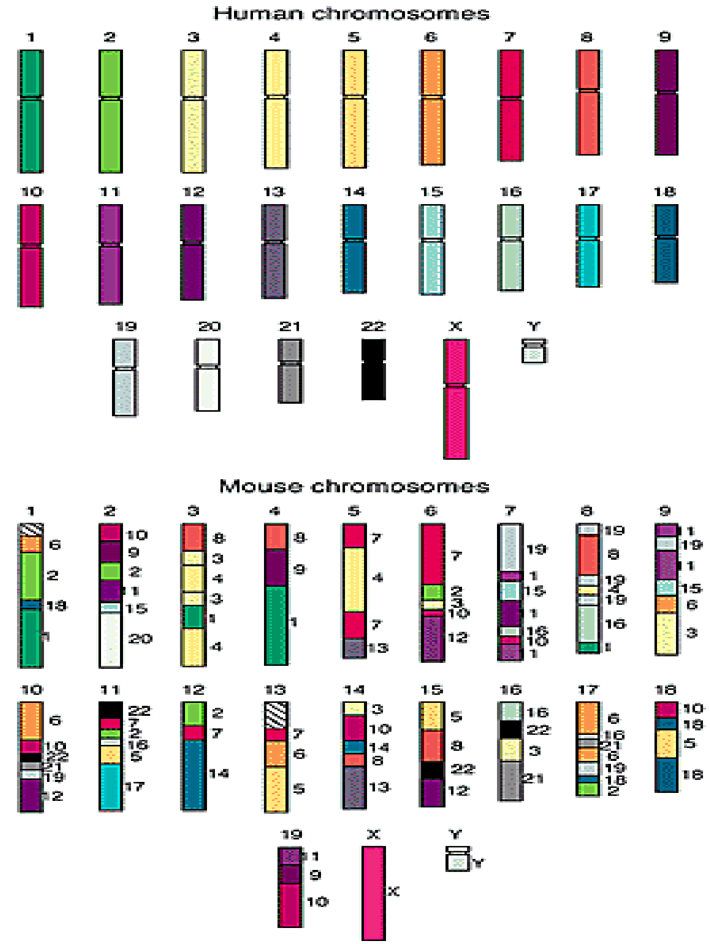
1 2 3 4 $\rightleftarrows$ 1 2 3 4 5 6

5 6

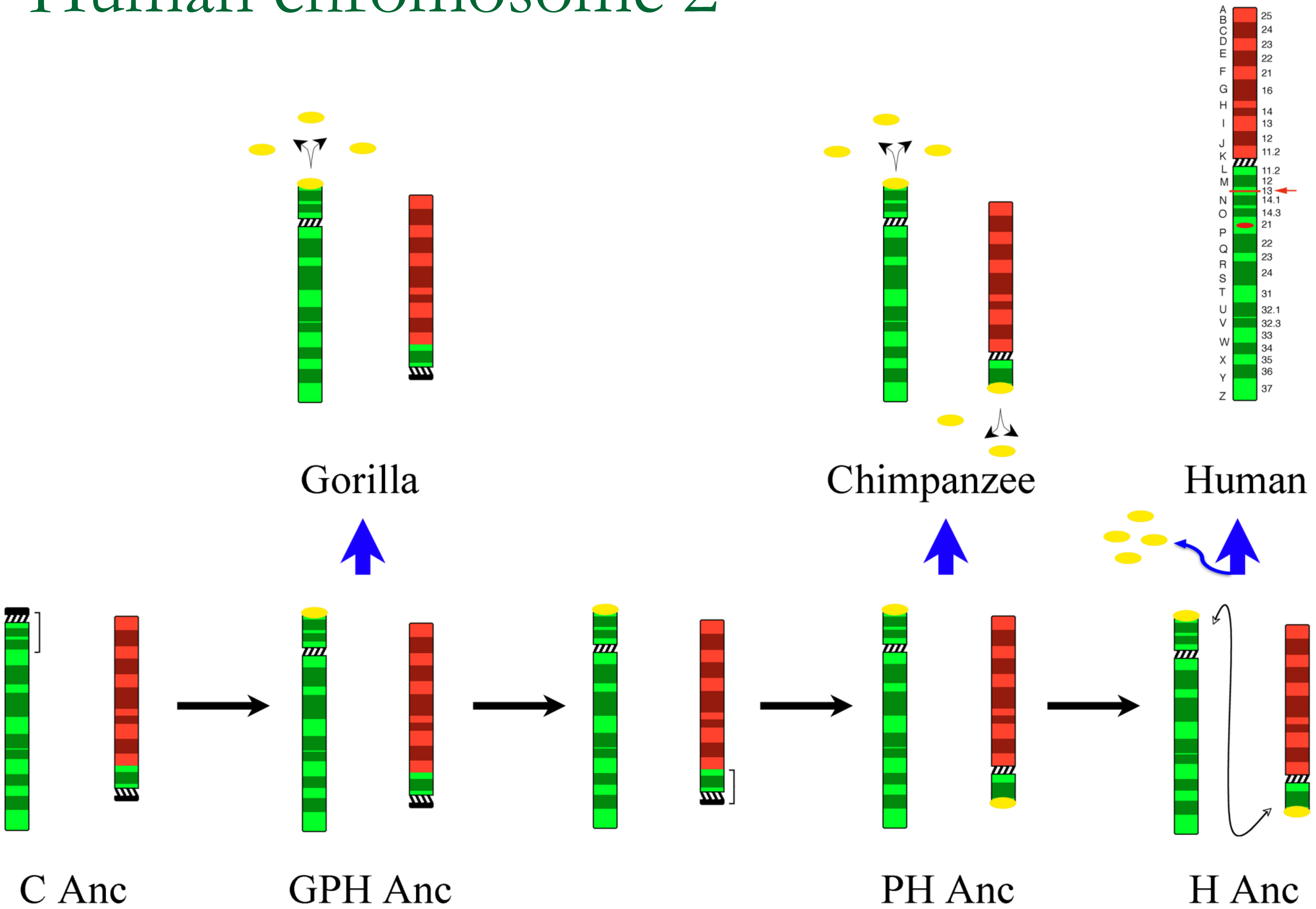
Fission

Comparative Genomic Architectures: Mouse vs Human Genome

- Humans and mice have similar genomes, but their genes are ordered differently
- ~245 rearrangements
  - Reversals
  - Fusions
  - Fissions
  - Translocation



Human chromosome 2



Reversals: Example

π = 1 2 3 4 5 6 7 8

$\rho(3,5)$

1 2 5 4 3 6 7 8

Reversals: Example

π = 1 2 <u>3 4 5</u> 6 7 8

$\rho(3,5)$ ↓

1 2 5 4 <u>3 6</u> 7 8

$\rho(5,6)$ ↓

1 2 5 4 6 3 7 8

Reversals and Gene Orders

- Gene order is represented by a permutation π:

$$\pi = \pi_1 \text{------} \pi_{i-1} \underline{\pi_i \, \pi_{i+1} \text{------} \pi_{j-1} \, \pi_j} \, \pi_{j+1} \text{-----} \pi_n$$

$$\Big\downarrow \rho(i,j)$$

$$\pi_1 \text{------} \pi_{i-1} \, \pi_j \, \pi_{j-1} \text{------} \pi_{i+1} \, \pi_i \, \pi_{j+1} \text{-----} \pi_n$$

- Reversal $\rho(i, j)$ reverses (flips) the elements from i to j in π

Reversal Distance Problem

- Goal: Given two permutations, find the shortest series of reversals that transforms one into another
- Input: Permutations $\boldsymbol{\pi}$ and $\boldsymbol{\sigma}$
- Output: A series of reversals $\rho_1,\ldots\rho_t$ transforming $\boldsymbol{\pi}$ into $\boldsymbol{\sigma}$, such that t is minimum
- ***t*** - reversal distance between π and σ
- ***d***(π, σ) - smallest possible value of t, given π and σ

Sorting By Reversals Problem

- Goal: Given a permutation, find a shortest series of reversals that transforms it into the identity permutation ($1\ 2\ \ldots\ n$)
- Input: Permutation π
- Output: A series of reversals $\rho_1, \ldots \rho_t$ transforming π into the identity permutation such that t is minimum

Sorting By Reversals: Example

- $t = d(\pi)$ - reversal distance of π
- Example :

$$
\begin{array}{cccccccccccc}
\pi & = & \underline{3} & \underline{4} & 2 & 1 & 5 & 6 & 7 & 10 & 9 & 8 \\
 & & 4 & 3 & 2 & 1 & 5 & 6 & 7 & \underline{10} & \underline{9} & \underline{8} \\
 & & \underline{4} & \underline{3} & \underline{2} & \underline{1} & 5 & 6 & 7 & 8 & 9 & 10 \\
 & & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10
\end{array}
$$

So $d(\pi) = 3$

Sorting by reversals: 5 steps

Step 0: π	2	-4	-3	5	-8	-7	-6	1
Step 1:	2	3	4	5	-8	-7	-6	1
Step 2:	2	3	4	5	6	7	8	1
Step 3:	2	3	4	5	6	7	8	-1
Step 4:	-8	-7	-6	-5	-4	-3	-2	-1
Step 5: γ	1	2	3	4	5	6	7	8

Sorting by reversals: 4 steps

Step 0: π	2	-4	-3	5	-8	-7	-6	1
Step 1:	2	3	4	5	-8	-7	-6	1
Step 2:	-5	-4	-3	-2	-8	-7	-6	1
Step 3:	-5	-4	-3	-2	-1	6	7	8
Step 4: γ	1	2	3	4	5	6	7	8

Pancake Flipping Problem

- The chef is sloppy; he prepares an unordered stack of pancakes of different sizes
- The waiter wants to rearrange them (so that the smallest winds up on top, and so on, down to the largest at the bottom)
- He does it by flipping over several from the top, repeating this as many times as necessary

Christos Papadimitrou and William H. Gates flip pancakes

Pancake Flipping Problem: Formulation

- Goal: Given a stack of n pancakes, what is the minimum number of flips to rearrange them into perfect stack?
- Input: Permutation π
- Output: A series of prefix reversals $\rho_1, \ldots \rho_t$ transforming π into the identity permutation such that t is minimum

Pancake Flipping Problem: Greedy Algorithm

- Greedy approach: 2 prefix reversals at most to place a pancake in its right position, $2n - 2$ steps total at most
- William Gates and Christos Papadimitriou showed in the mid-1970s that this problem can be solved by at most *5/3 (n + 1) prefix reversals*

Sorting By Reversals: A Greedy Algorithm

- If sorting permutation π = 1 2 3 6 4 5, the first three elements are already in order so it does not make any sense to break them.
- The length of the already sorted prefix of π is denoted *prefix*(π)
  - *prefix*(π) = 3
- This results in an idea for a greedy algorithm: increase *prefix*(π) at every step

Greedy Algorithm: An Example

- Doing so, π can be sorted

$$\color{red}{1\ 2\ 3}\ \underline{6\ 4}\ 5$$

$$\downarrow$$

$$\color{red}{1\ 2\ 3\ 4}\ \underline{6\ 5}$$

$$\downarrow$$

$$\color{red}{1\ 2\ 3\ 4\ 5\ 6}$$

- Number of steps to sort permutation of length *n* is at most *(n – 1)*

Greedy Algorithm: Pseudocode

```
SimpleReversalSort(π)
for i ← 1 to n – 1
   j ← position of element i in π (i.e., π_j = i)
   if j ≠ i
      π ← π * ρ(i, j)
      output π
   if π is the identity permutation
     return
```

Analyzing SimpleReversalSort

- SimpleReversalSort does not guarantee the smallest number of reversals and takes five steps on $\pi = 6\ 1\ 2\ 3\ 4\ 5$:
  - Step 1: 1 6 2 3 4 5
  - Step 2: 1 2 6 3 4 5
  - Step 3: 1 2 3 6 4 5
  - Step 4: 1 2 3 4 6 5
  - Step 5: 1 2 3 4 5 6

Analyzing SimpleReversalSort (cont'd)

- But it can be sorted in two steps:

  π = 6 1 2 3 4 5

  - Step 1: 5 4 3 2 1 6
  - Step 2: 1 2 3 4 5 6
- So, SimpleReversalSort(π) is not optimal
- Optimal poly-time algorithms are unknown for NP-hard problems; approximation algorithms are used

Approximation Algorithms

- These algorithms find approximate solutions rather than optimal solutions
- The approximation ratio of an algorithm A on input π is:

$$A(\pi) / OPT(\pi)$$

where

$A(\pi)$ - solution produced by algorithm A

$OPT(\pi)$ - optimal solution of the problem

Approximation Ratio/Performance Guarantee

- Approximation ratio (performance guarantee) of algorithm A: max approximation ratio of all inputs of size *n*
  - For algorithm A that minimizes objective function (minimization algorithm):
    - $\max_{|\pi| = n} A(\pi) \,/\, OPT(\pi)$

Approximation Ratio/Performance Guarantee

- Approximation ratio (performance guarantee) of algorithm A: max approximation ratio of all inputs of size *n*
  - For algorithm A that minimizes objective function (minimization algorithm):
    - $\max_{|\pi| = n} A(\pi) \ / \ OPT(\pi)$
  - For maximization algorithm:
    - $\min_{|\pi| = n} A(\pi) \ / \ OPT(\pi)$

Adjacencies and Breakpoints

$$\pi = \pi_1\pi_2\pi_3\ldots\pi_{n-1}\pi_n$$

- A pair of elements π_i and π_{i+1} are adjacent if

$$\pi_{i+1} = \pi_i \pm 1$$

- For example:

$$\pi = 1 \;\; 9 \;\; \underline{3 \;\; 4} \;\; \underline{7 \;\; 8} \;\; 2 \;\; \underline{6 \;\; 5}$$

- (3, 4) or (7, 8) and (6,5) are adjacent pairs

Breakpoints

There is a breakpoint between any adjacent element that are non-consecutive:

$$\pi = 1 \mid 9 \mid 3 \;\; 4 \mid 7 \;\; 8 \mid 2 \mid 6 \;\; 5$$

- Pairs (1,9), (9,3), (4,7), (8,2) and (2,6) form breakpoints of permutation π
- $b(\pi)$ - # breakpoints in permutation π

Adjacency & Breakpoints

- An **adjacency** - a pair of adjacent elements that are **consecutive**
- A **breakpoint** - a pair of adjacent elements that are **not consecutive**

$\pi = 5\ \ 6\ \ 2\ \ 1\ \ 3\ \ 4 \longrightarrow$ **Extend** π **with** $\pi_0 = 0$ **and** $\pi_7 = 7$

adjacencies

0 5 6 2 1 3 4 7

breakpoints

Extending Permutations

- We put two elements $\pi_0 = 0$ and $\pi_{n+1} = n+1$ at the ends of π

  Example:

$$\pi = 1 \mid 9 \mid 3 \;\; 4 \mid 7 \;\; 8 \mid 2 \mid 6 \;\; 5$$

Extending with *0* and *10*

$$\pi = 0 \; 1 \mid 9 \mid 3 \;\; 4 \mid 7 \;\; 8 \mid 2 \mid 6 \;\; 5 \mid 10$$

Note: A new breakpoint was created after extending

Reversal Distance and Breakpoints

- Each reversal eliminates at most 2 breakpoints.

π = 2 3 1 4 6 5

$0 \mid \underline{2\ \ 3 \mid 1} \mid 4 \mid 6\ \ 5 \mid 7$ $\qquad b(\pi) = 5$

$0\ \ 1 \mid \underline{3\ \ 2} \mid 4 \mid 6\ \ 5 \mid 7$ $\qquad b(\pi) = 4$

$0\ \ 1\ \ 2\ \ 3\ \ 4 \mid \underline{6\ \ 5} \mid 7$ $\qquad b(\pi) = 2$

$0\ \ 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6\ \ 7$ $\qquad b(\pi) = 0$

Reversal Distance and Breakpoints

- Each reversal eliminates at most 2 breakpoints.
- This implies:

  reversal distance ≥ #breakpoints / 2

π = 2 3 1 4 6 5

$0 \mid \underline{2\ \ 3 \mid 1} \mid 4 \mid 6\ \ 5 \mid 7$ $\qquad b(\pi) = 5$

$0\ \ 1 \mid \underline{3\ \ 2} \mid 4 \mid 6\ \ 5 \mid 7$ $\qquad b(\pi) = 4$

$0\ \ 1\ \ 2\ \ 3\ \ 4 \mid \underline{6\ \ 5} \mid 7$ $\qquad b(\pi) = 2$

$0\ \ 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6\ \ 7$ $\qquad b(\pi) = 0$

Sorting By Reversals: A Better Greedy Algorithm

```
BreakPointReversalSort(π)
while b(π) > 0
   Among all possible reversals,
   choose reversal ρ minimizing b(π • ρ)
   π ← π • ρ(i, j)
   output π
return
```

Sorting By Reversals: A Better Greedy Algorithm

<u>BreakPointReversalSort(π)</u>

while $b(\pi) > 0$

  Among all possible reversals, choose reversal ρ minimizing $b(\pi \cdot \rho)$

  $\pi \leftarrow \pi \cdot \rho(i, j)$

  output π

return

Problem: this algorithm may work forever

Strips

- Strip: an interval between two consecutive breakpoints in a permutation
  - Decreasing strip: *strip* of elements in decreasing order (e.g. 6 5 and 3 2).
  - Increasing strip: *strip* of elements in increasing order (e.g. 7 8)

0 1 9 4 3 7 8 2 5 6 *10*

  - A single-element strip can be declared either increasing or decreasing. We will choose to declare them as decreasing with exception of the strips with *0* and *n+1*

Reducing the Number of Breakpoints

Theorem 1:

If permutation π contains at least one decreasing strip, then there exists a reversal ρ which decreases the number of breakpoints (i.e. $b(\pi \bullet \rho) < b(\pi)$)

Things To Consider

- For π = 1 4 6 5 7 8 3 2

  $0 \; 1 \,|\, 4 \,|\, 6 \; 5 \,|\, 7 \; 8 \,|\, 3 \; 2 \,|\, 9$ $\quad b(\pi) = 5$

  - Choose decreasing strip with the smallest element k in π (k = 2 in this case)

Things To Consider (cont'd)

- For π = 1 4 6 5 7 8 3 2

  $0 \ 1 \,|\, 4 \,|\, 6 \ 5 \,|\, 7 \ 8 \,|\, 3 \ 2 \,|\, 9$ $\quad b(\pi) = 5$

  - Choose decreasing strip with the smallest element k in π (k = 2 in this case)

Things To Consider (cont'd)

- For π = 1 4 6 5 7 8 3 2

  $0\ \ 1\ |\ 4\ |\ 6\ \ 5\ |\ 7\ \ 8\ |\ 3\ \ 2\ |\ 9$ $\qquad b(\pi) = 5$

  - Choose decreasing strip with the smallest element *k* in π (*k* = 2 in this case)
  - Find *k – 1* in the permutation

Things To Consider (cont'd)

- For π = 1 4 6 5 7 8 3 2

  $0 \;\; 1 \,|\, 4 \,|\, 6 \;\; 5 \,|\, 7 \;\; 8 \,|\, 3 \;\; 2 \,|\, 9$ $\qquad b(\pi) = 5$

  - Choose decreasing strip with the smallest element *k* in π (*k* = 2 in this case)
  - Find *k – 1* in the permutation
  - Reverse the segment between *k* and *k-1*:
  - $0 \;\; 1 \,|$ [redacted] $|\, 6 \;\; 5 \,|\, 7 \;\; 8 \,|\, 3 \;\; 2 \,|\, 9$ $\qquad b(\pi) = 5$
  - $0 \;\; 1 \;\; 2 \;\; 3 \,|\, 8 \;\; 7 \,|\, 5 \;\; 6 \,|\, 4 \,|\, 9$ $\qquad b(\pi) = 4$

Reducing the Number of Breakpoints Again

- If there is no decreasing strip, there may be no reversal ρ that reduces the number of breakpoints (i.e. $b(\pi \bullet \rho) \geq b(\pi)$ for any reversal ρ).
- By reversing an increasing strip (# of breakpoints stay unchanged), we will create a decreasing strip at the next step. Then the number of breakpoints will be reduced in the next step (theorem 1).

Things To Consider (cont'd)

- There are no decreasing strips in π, for:

$$\pi = \mathit{0}\ 1\ 2\ |\ 5\ 6\ 7\ |\ 3\ 4\ |\ \mathit{8} \quad b(\pi) = 3$$

$$\pi \bullet \rho(6,7) = \mathit{0}\ 1\ 2\ |\ 5\ 6\ 7\ |\ 4\ 3\ |\ \mathit{8} \quad b(\pi) = 3$$

- ✓ $\rho(6,7)$ does not change the # of breakpoints
- ✓ $\rho(6,7)$ creates a decreasing strip thus guaranteeing that the next step will decrease the # of breakpoints.

ImprovedBreakpointReversalSort

<u>ImprovedBreakpointReversalSort(π)</u>

while $b(\pi) > 0$

if π has a decreasing strip

Among all possible reversals, choose reversal ρ that minimizes $b(\pi \bullet \rho)$

else

Choose a reversal ρ that flips an increasing strip in π

$\pi \leftarrow \pi \bullet \rho$

output π

return

ImprovedBreakpointReversalSort: Performance Guarantee

- *ImprovedBreakPointReversalSort* is an approximation algorithm with a performance guarantee of at most 4
  - It eliminates at least one breakpoint in every two steps; at most $2b(\pi)$ steps
  - Approximation ratio: $2b(\pi) \; / \; d(\pi)$
  - Optimal algorithm eliminates at most 2 breakpoints in every step: $d(\pi) \geq b(\pi) / 2$
  - Performance guarantee:
    - $(\; 2b(\pi) \; / \; d(\pi) \;) \geq [\; 2b(\pi) \; / \; (b(\pi) \; / \; 2) \;] = 4$