Computational Genomics
Prof. Ron Shamir & Prof. Roded Sharan
School of Computer Science, Tel Aviv University

גנומיקה חישובית
פרופ' רון שמיר ופרופ' רודד שרן
ביה"ס למדעי המחשב,אוניברסיטת תל אביב

Lecture 13: Genome Rearrangements

11/1/13

Genome Rearrangements

Slides with Itsik Pe'er, Michal Ozery-Flato, Tamar Barzuza

Additional sources:

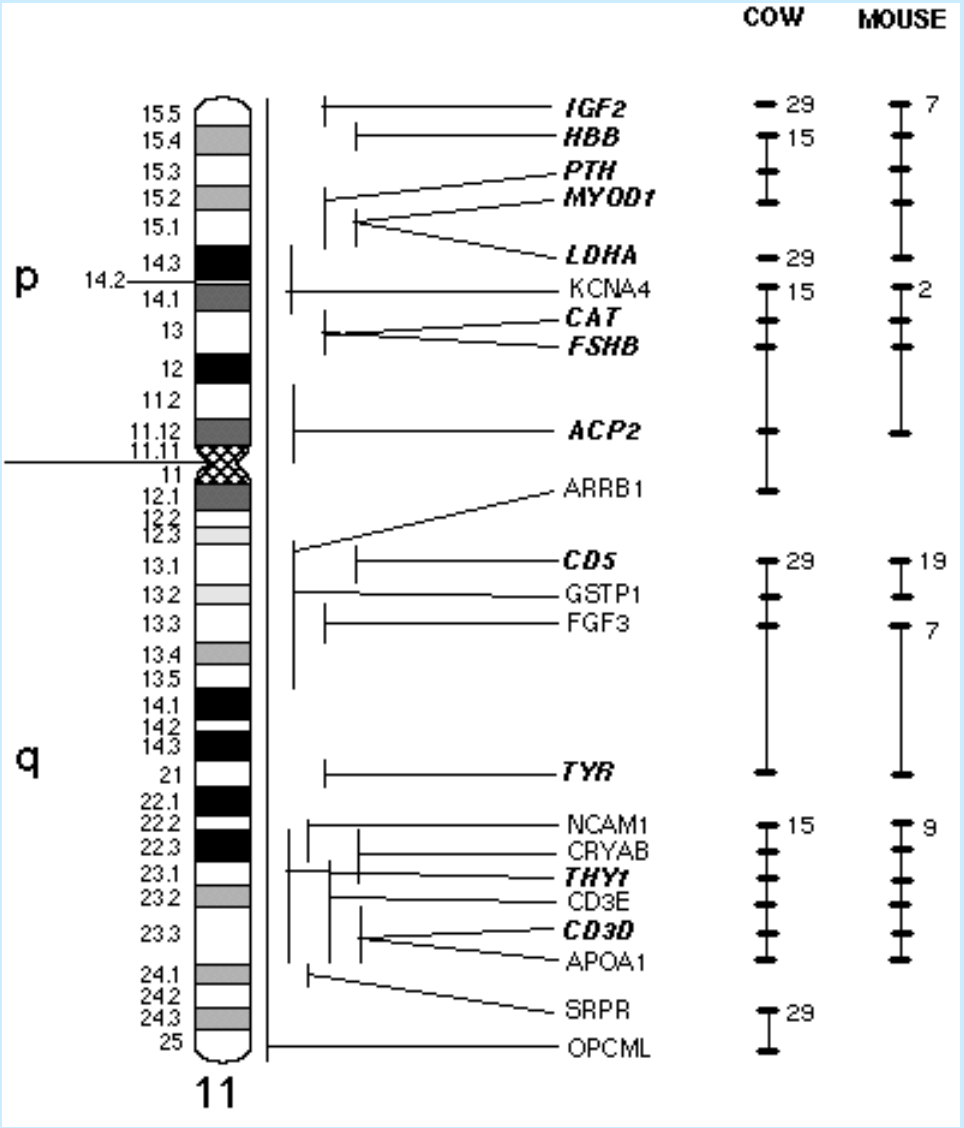
- E. Tannier's CPM'04 slides
- V. Helms Bioinfo III course (Saarlands)
- P.A. Pevzner, N. Jones BioAlgorithms course www.bioalgorithms.info

(a)

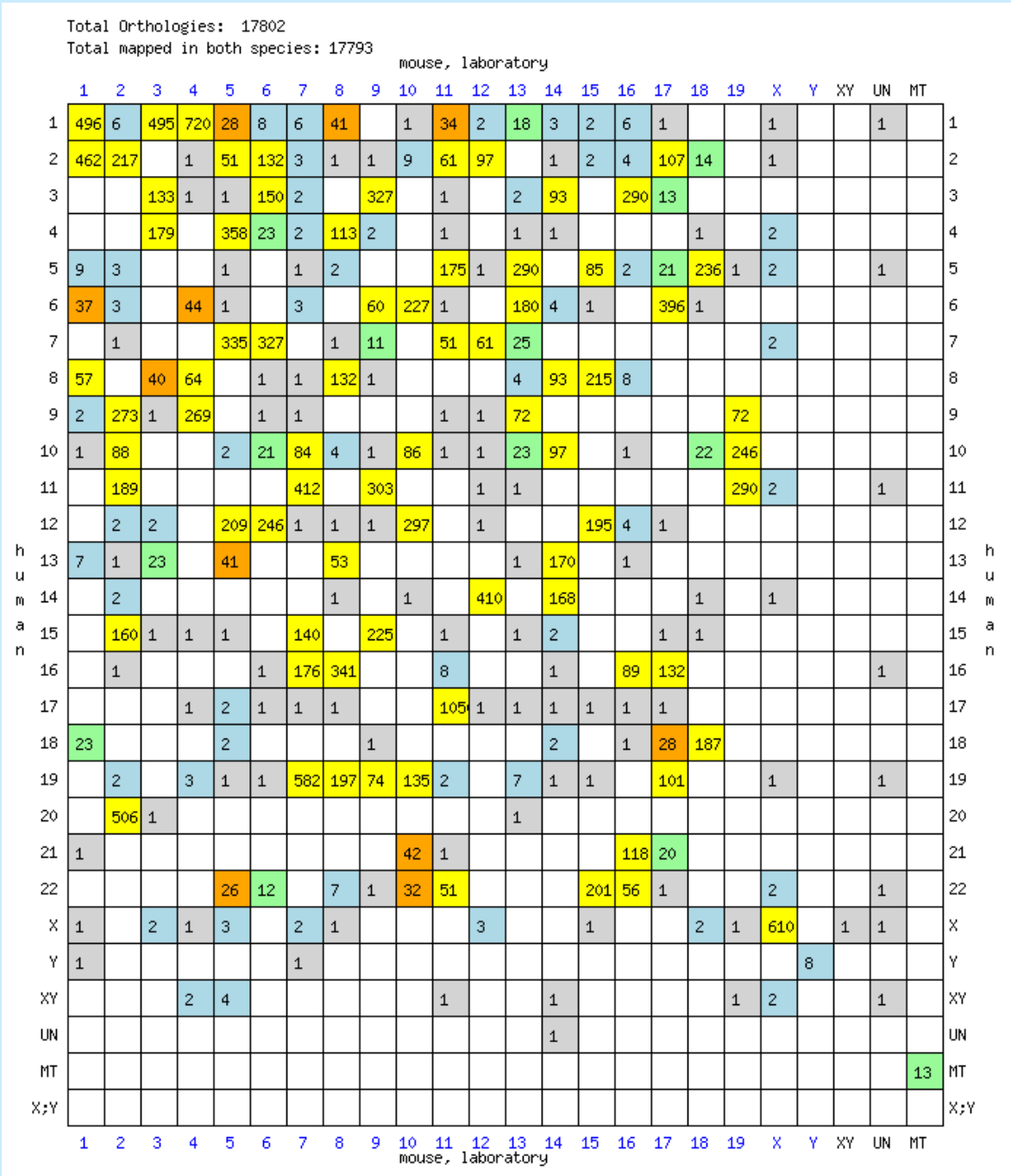
(b) 5μm

Figure 1-17
(a) Electron micrograph of chromosome 4 from a salivary gland of *Chironomus tentans*. [Reproduced with permission from B. Daneholt, *Cell* 4 (1975):1.] (b) Diagram of a portion of a salivary gland chromosome (the right arm of chromosome 3) of *Drosophila melanogaster*. [After P. N. Bridges, *J. Heredity* 32 (1941).]

Comparative map: Human Chr 11 vs cow, mouse (12/00)



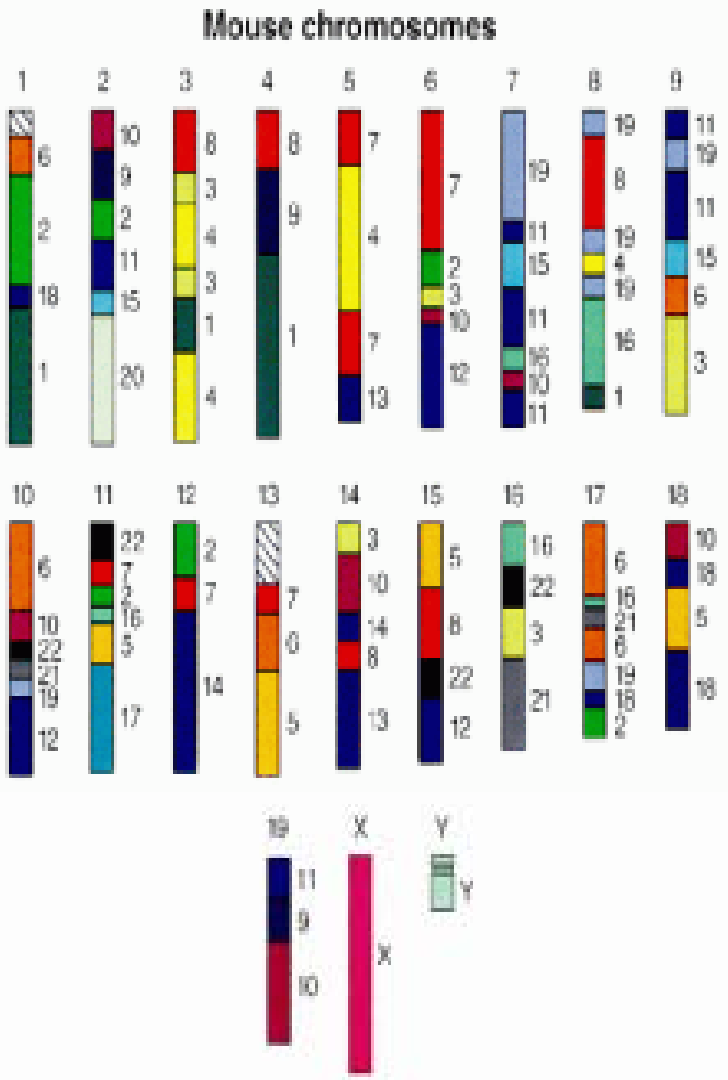
http://bos.cvm.tamu.edu/

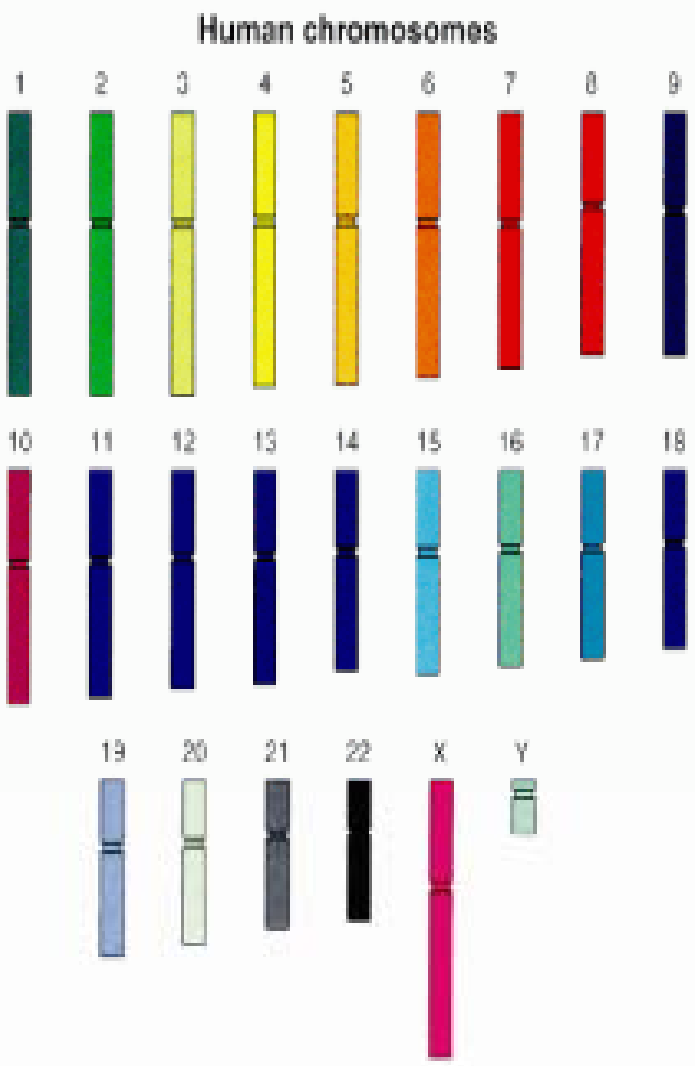


Oxford Grid: human vs mouse

(1/2010)

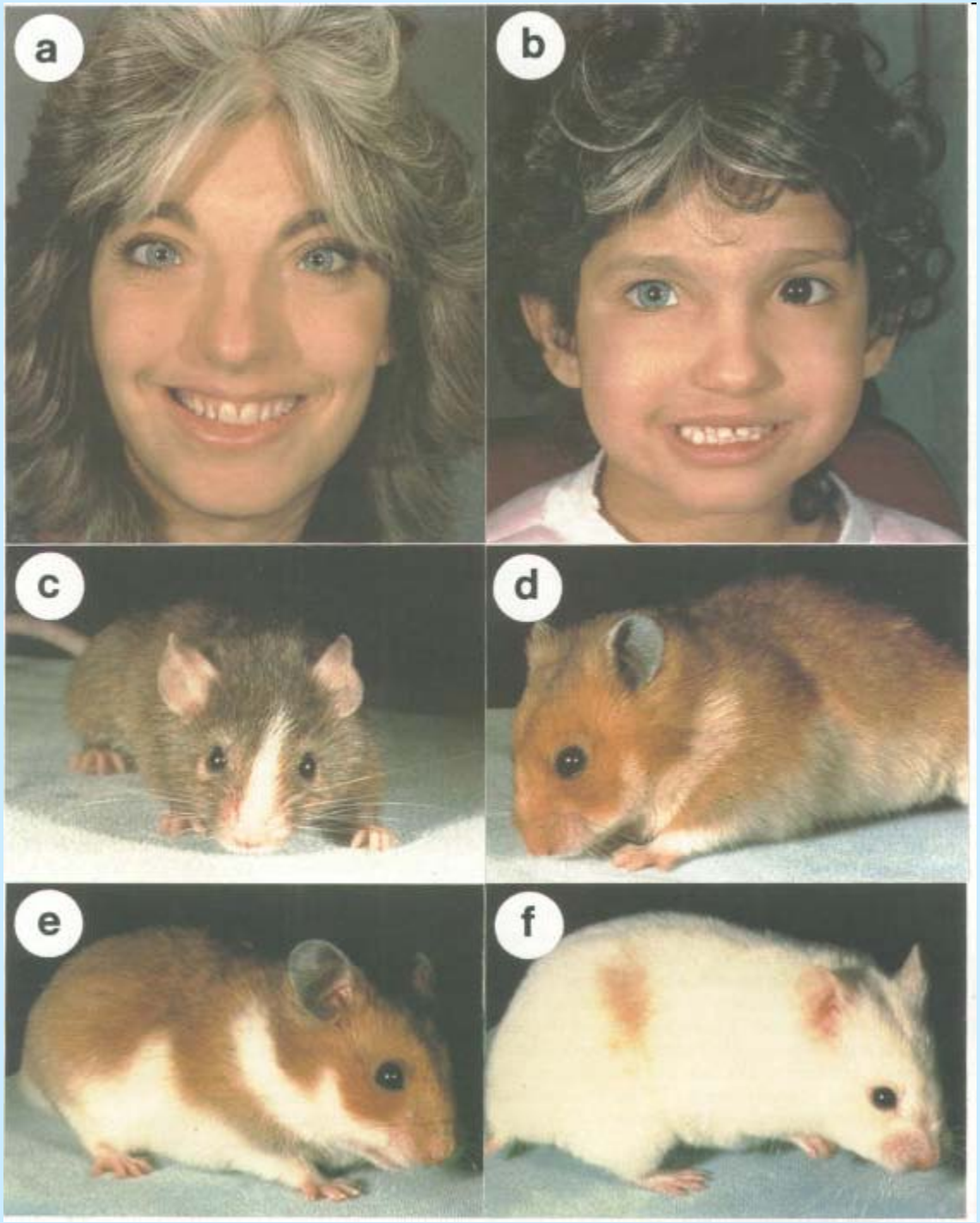
http://www.informatics.jax.org/searches/oxfordgrid_form.shtml


Mouse chromosomes
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
X
Y


Human chromosomes
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
X
Y

Waardenburg's Syndrome: Mouse Provides Insight into Human Genetic Disorder

- Waardenburg's syndrome is characterized by hearing loss, neurological problems and pigmentary dysphasia
- Gene implicated in the disease was linked to human chromosome 2 but it was not clear where exactly it is located on chromosome 2



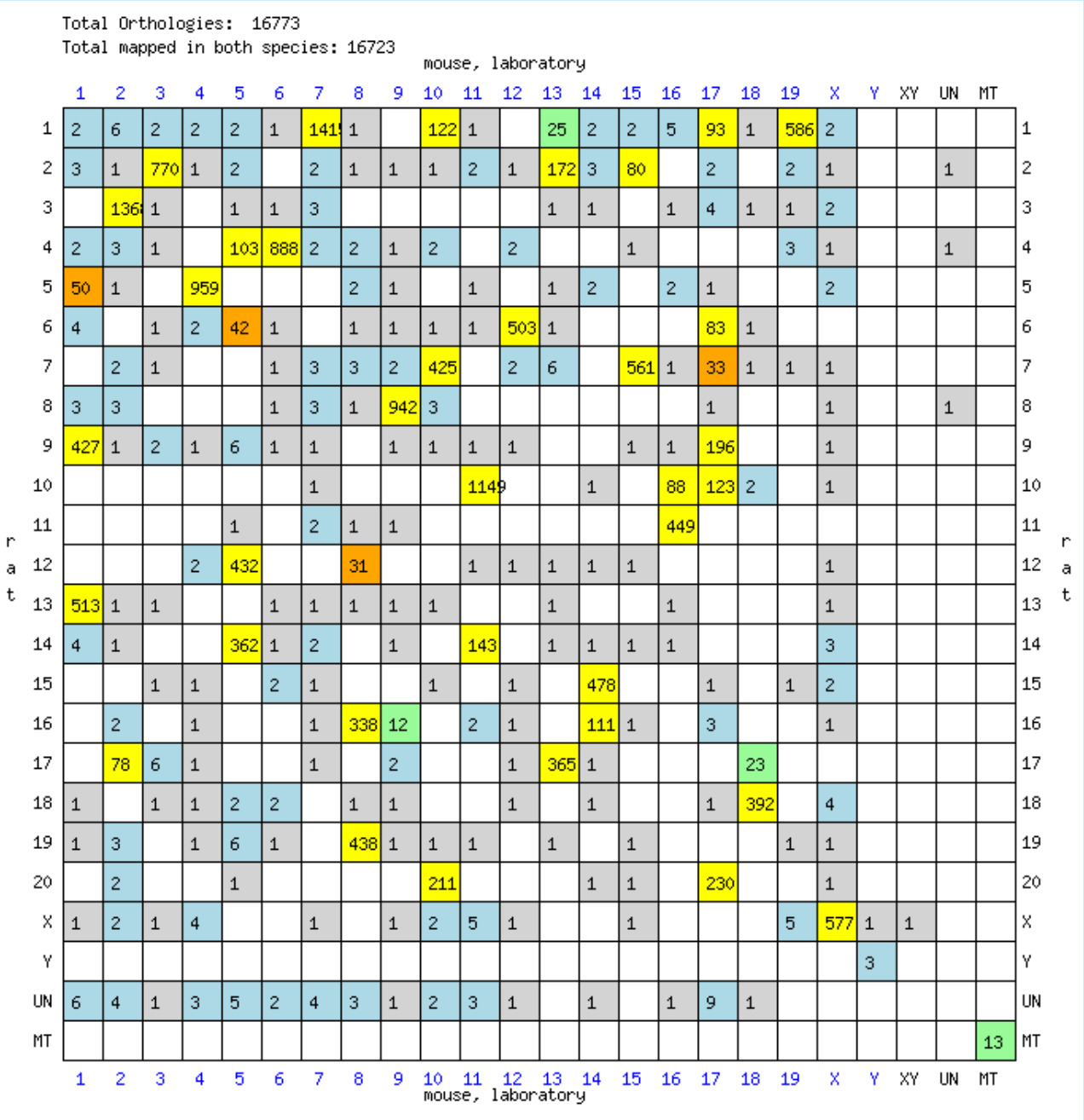
http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1017240/

Waardenburg's syndrome and splotch mice

- A breed of mice (with splotch gene) had similar symptoms caused by the same type of gene as in humans

- Scientists succeeded in identifying location of gene responsible for disorder in mice

- Finding the gene in mice gives clues to where the same gene is located in humans

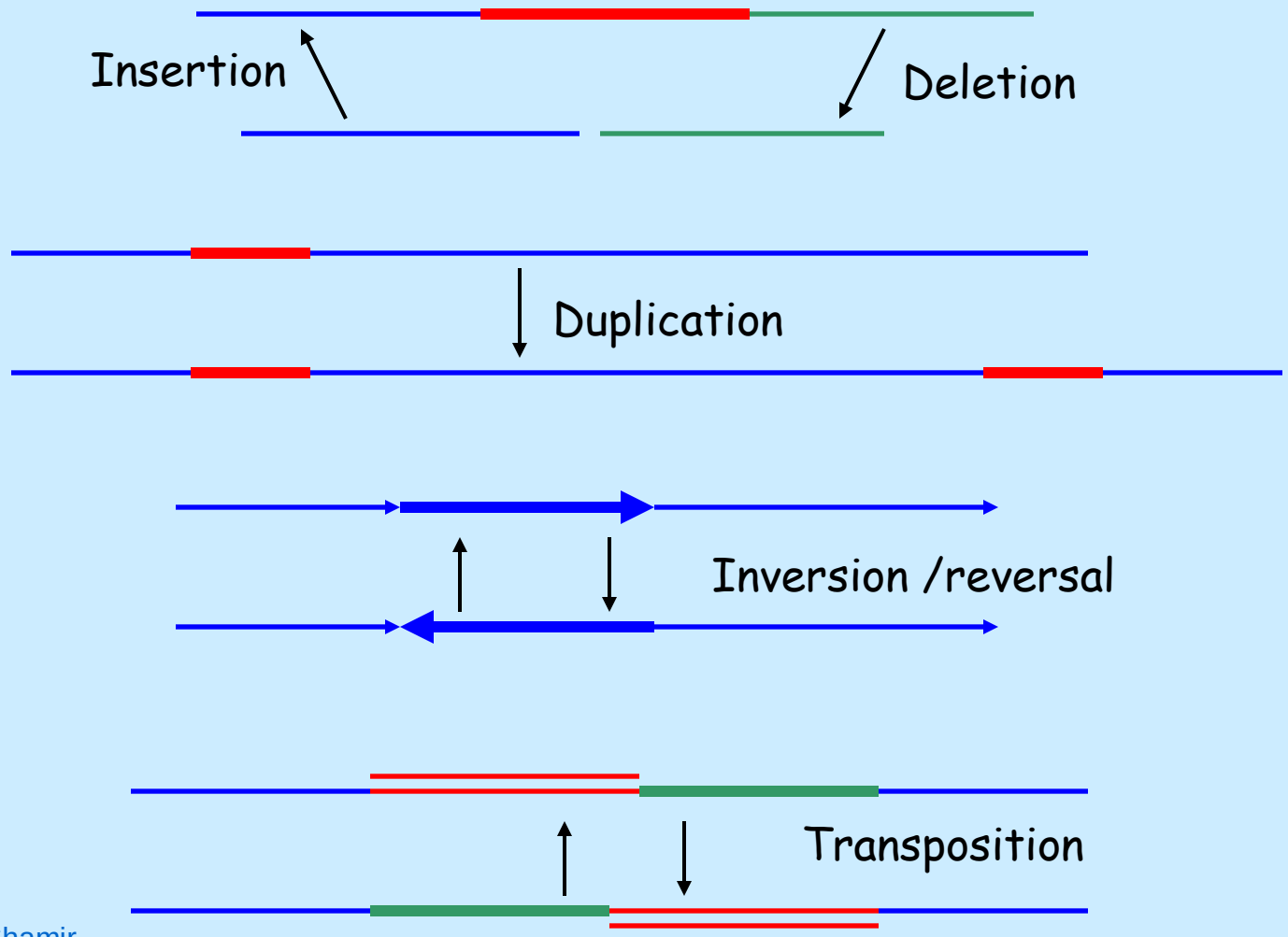


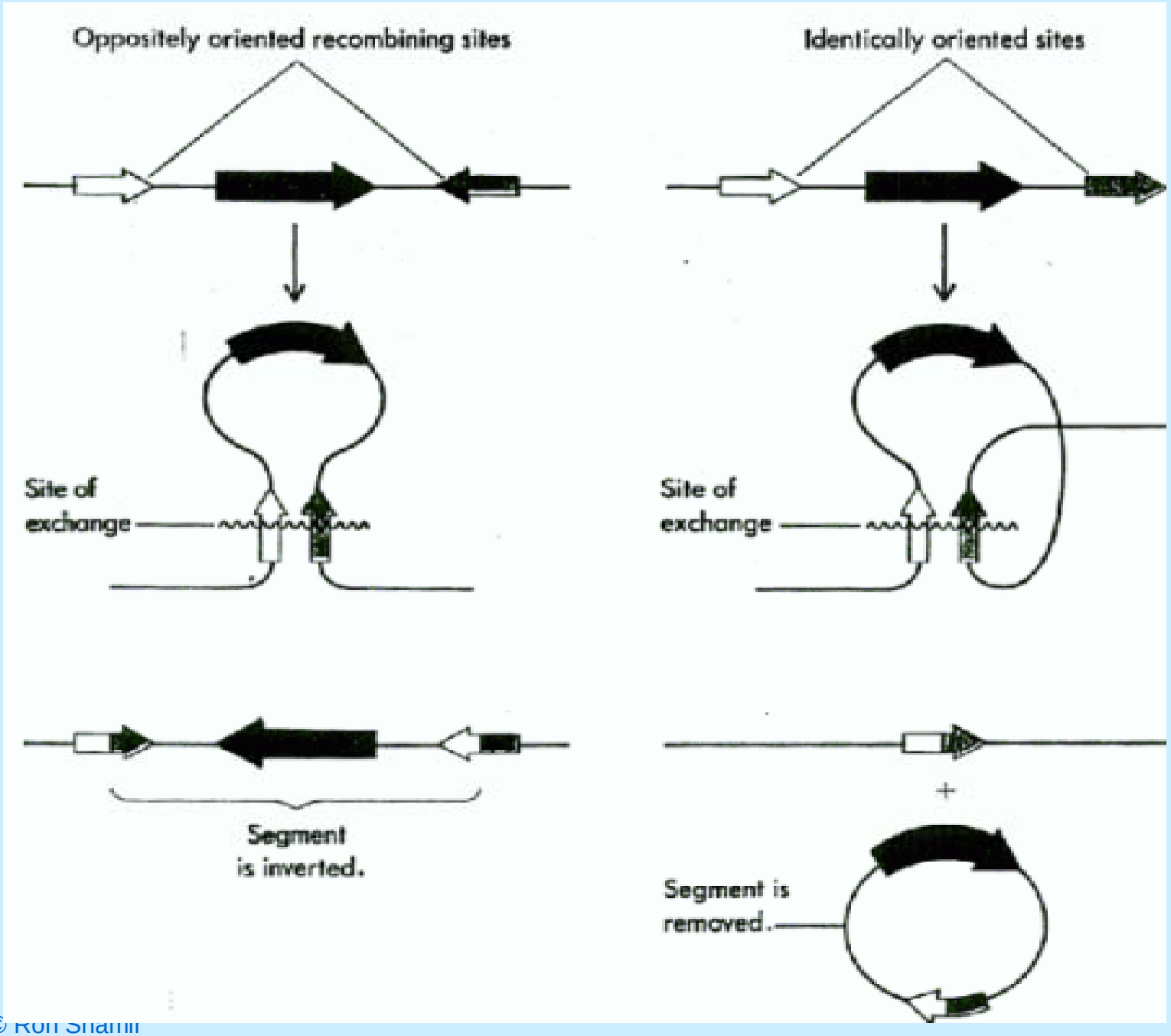
Oxford Grid: rat vs mouse

(1/2010)

Genomic Rearrangements (GR)

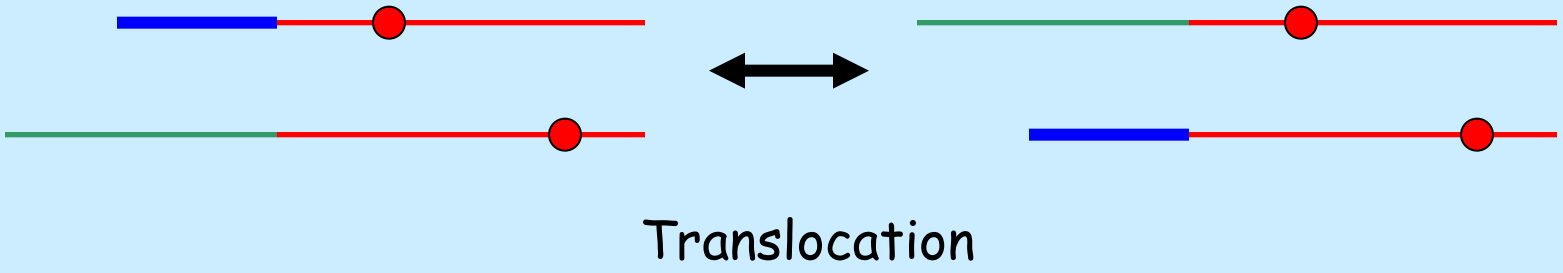
- Single Chromosome:



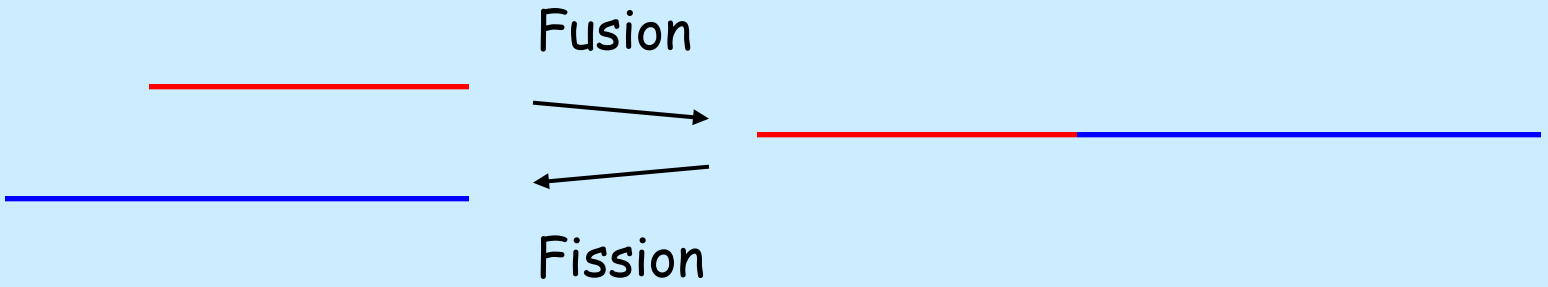

Oppositely oriented recombining sites
Identically oriented sites
Site of exchange
Site of exchange
Segment is inverted.
Segment is removed.

Genomic Rearrangements (GR)

- Inter – Chromosome:



Translocation



Why study GR?

Evolution!

- Rare events – can allow phylogenetic inference much further back
- Less ambiguity than on base level
- Larger scale data: chromosome, genome
- Better multi-species analysis

BG…

Reversals

Assume: All genes on chromosome are distinguishable

⟶ Transform to permutation

Π_1	1	2	3	4	5	6
	1	4	3	2	5	6
	1	4	6	5	2	3
Π_2	6	4	1	5	2	3

Goal: Given π, find its *reversal distance* from id

Kececioglu-Sankoff	95	2-approx, b&b
Bafna-Pevzner	96	1.75-approx
Caprara	97	NPC
Christie	98	1.5-approx
Berman, Karpinski	99	MAX-SNP hard
Berman, Hannenhalli, Karpinski	01	1.375-approx

Breakpoint in π: $|\pi_i - \pi_{i+1}| \neq 1$

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

d d d d i d

Strips
i: increasing>1
d: decreasing≥1

$b(\pi)$:= #bp in π

Δb := change in #bp in a step

$d(\pi)$:= reversal distance of π

Observation: OPT=$d(\pi) \geq \lceil b(\pi)/2 \rceil$

Lemma: if π contains a decreasing strip, there is a reversal that decreases #bp by ≥ 1

key: use decreasing strip with smallest element

"good"

Alg: If ∃ decr. strip, find and perform good reversal $\Delta b=-1$

Else reverse an inc. strip $\Delta b=0$

Performance: $\leq$ 2b inversions $\leq 4 \cdot OPT$

Lemma (Kececioglu – Sankoff '95) : If ∄ reversal with $\Delta b=-1$ that leaves a decreasing strip, then ∃ a reversal with $\Delta b=-2$

➔ New approximation alg with ≤ 2·OPT reversals:

- As long as possible:
  - reverse a good decreasing strip, leaving a decreasing strip

$\Delta b = -1$ in one step

- if impossible:
  - do a reversal with $\Delta b=-2$
  - reverse any strip

$\Delta b = -2$ in two steps

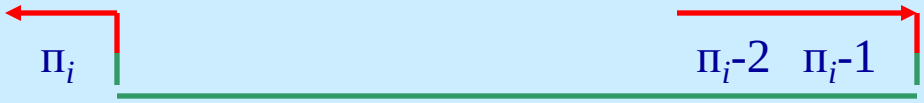
Lemma: (Kececioglu – Sankoff '95)

If every reversal that removes a breakpoint leaves a permutation without decreasing strip, then π has a reversal that removes two breakpoints

Proof: π_i – smallest element in decreasing strip

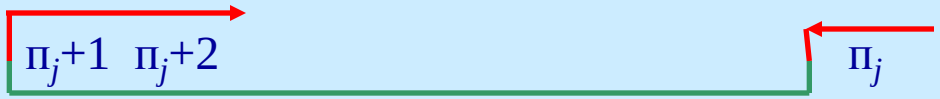
π_j – greatest element in decreasing strip

(1)



impossible (rev. leaves a decr. strip)

(2)



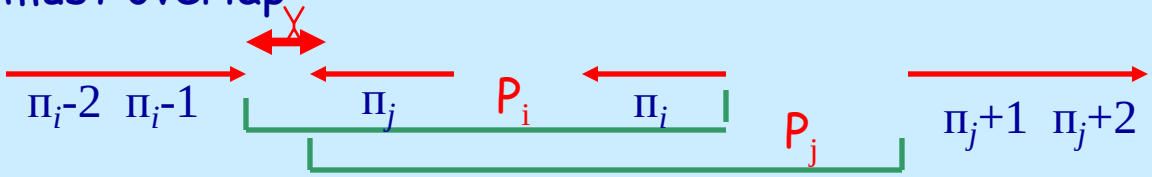
impossible

➔

situation:

(3) P_i, P_j must overlap

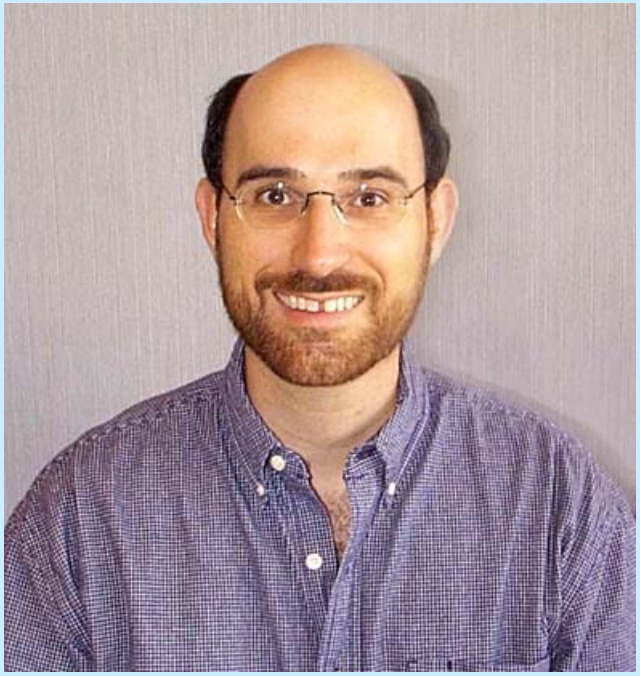
⇒

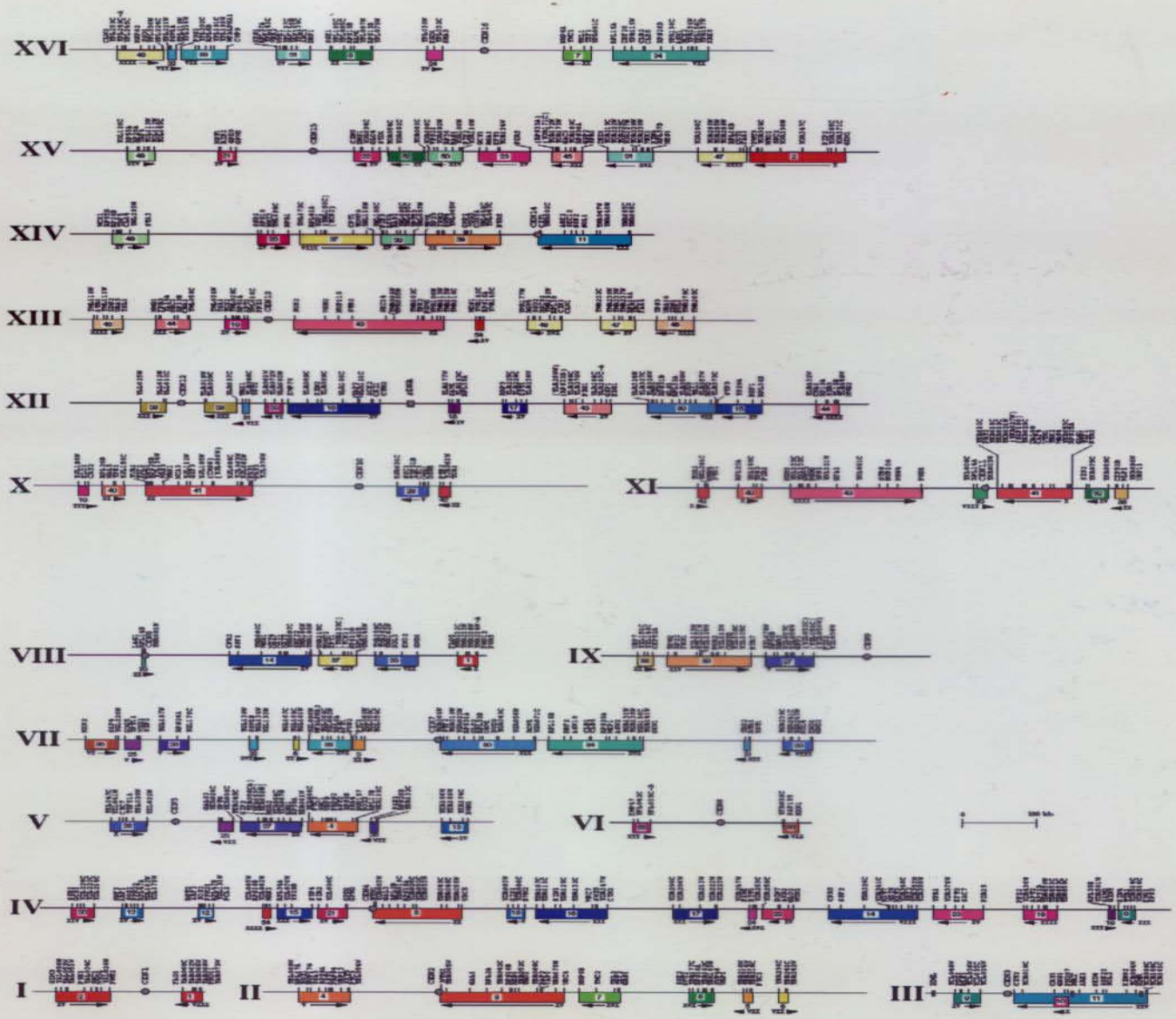


If $P_i \setminus P_j \neq \varnothing$ contains decreasing strip – apply P_j
increasing strip – apply P_i } ➔ $P_i \setminus P_j = \varnothing$

Similarly $P_j \setminus P_i = \varnothing$ ➔ $P_i = P_j$ ➔ 2 breakpoints!

David Sankoff, John Kececioglou




XVI
XV
XIV
XIII
XII
X
XI
VIII
IX
VII
V
VI
IV
I
II
III

Sorting signed permutations by reversals

Sorting by Reversals (SBR)

0 7 5 3 -1 -6 -2 4 8 (HS)

0 1 2 3 4 5 6 7 8 (MM)

Sorting by Reversals

0 7 5 3 -1 -6 -2 4 8 (HS)

0 1 -3 -5 -7 -6 -2 4 8

0 1 2 3 4 5 6 7 8 (MM)

Sorting by Reversals

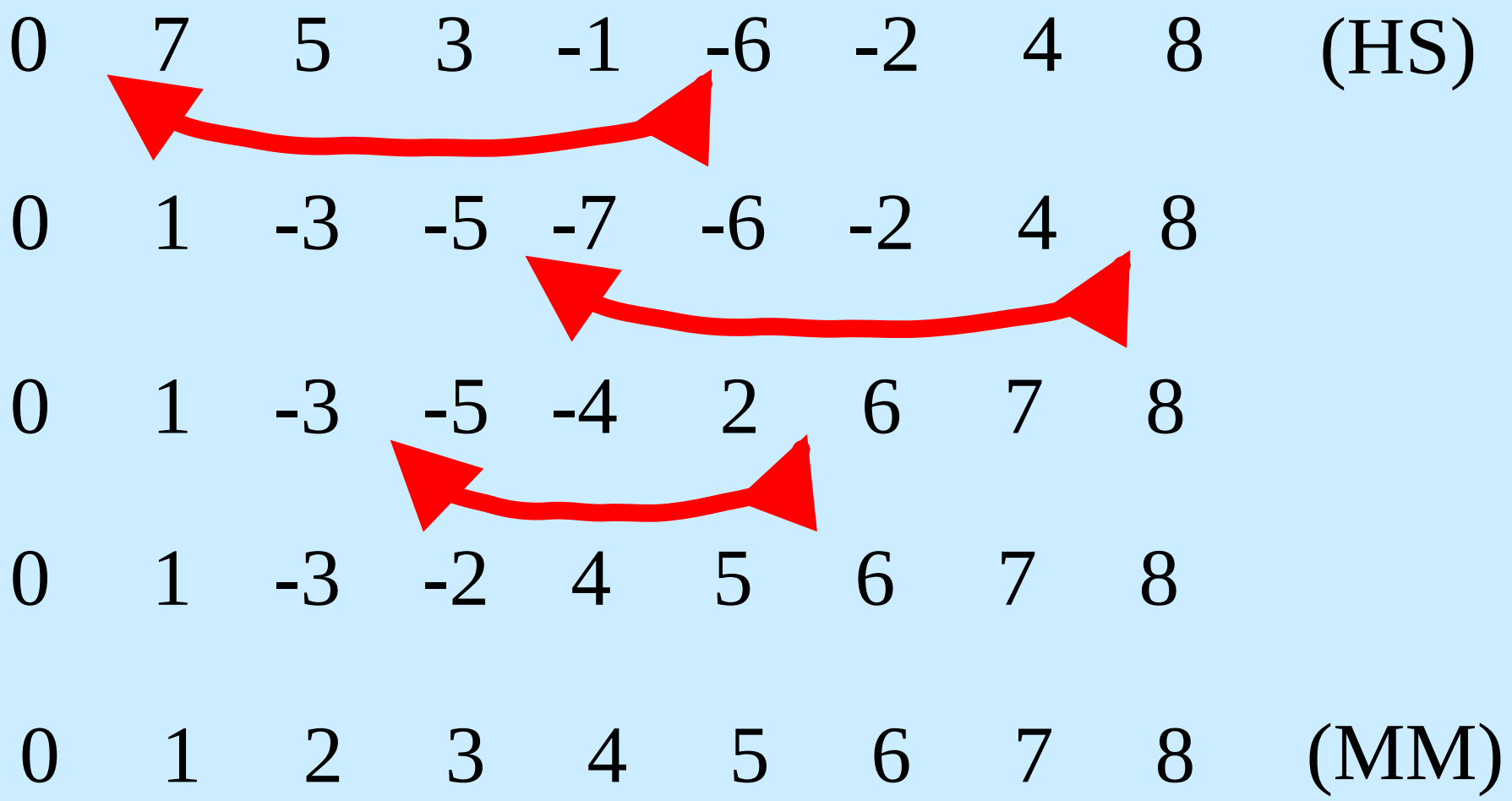
0 7 5 3 -1 -6 -2 4 8 (HS)

0 1 -3 -5 -7 -6 -2 4 8

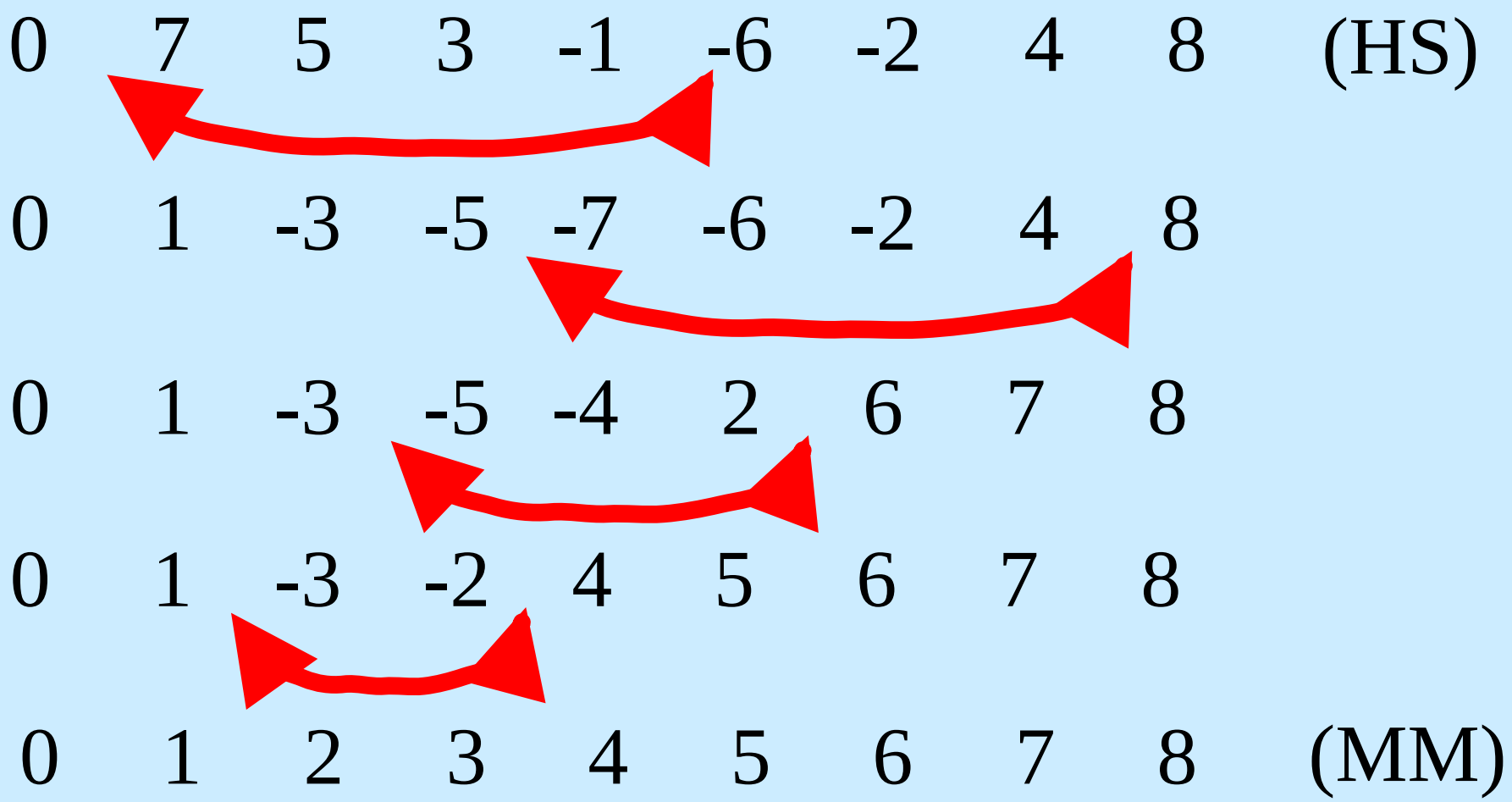
0 1 -3 -5 -4 2 6 7 8

0 1 2 3 4 5 6 7 8 (MM)

Sorting by Reversals



Sorting by Reversals



A *Signed Permutation*:

4 -3 1 -5 -2 7 6

Reversal r(i,j):

Flip order, signs of numbers in positions i,i+1,..j

After r(4,6):

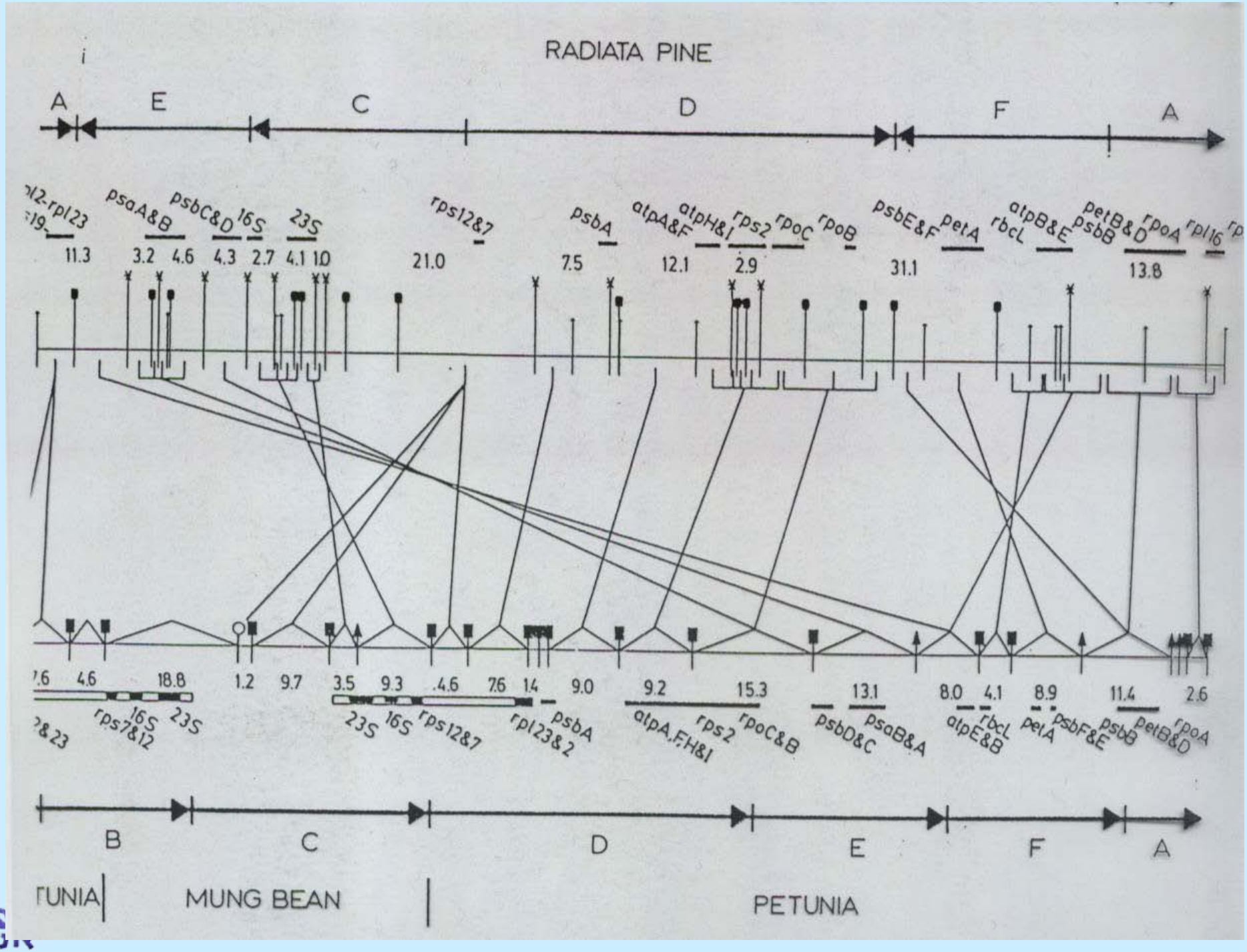
4 -3 1 -7 2 5 6

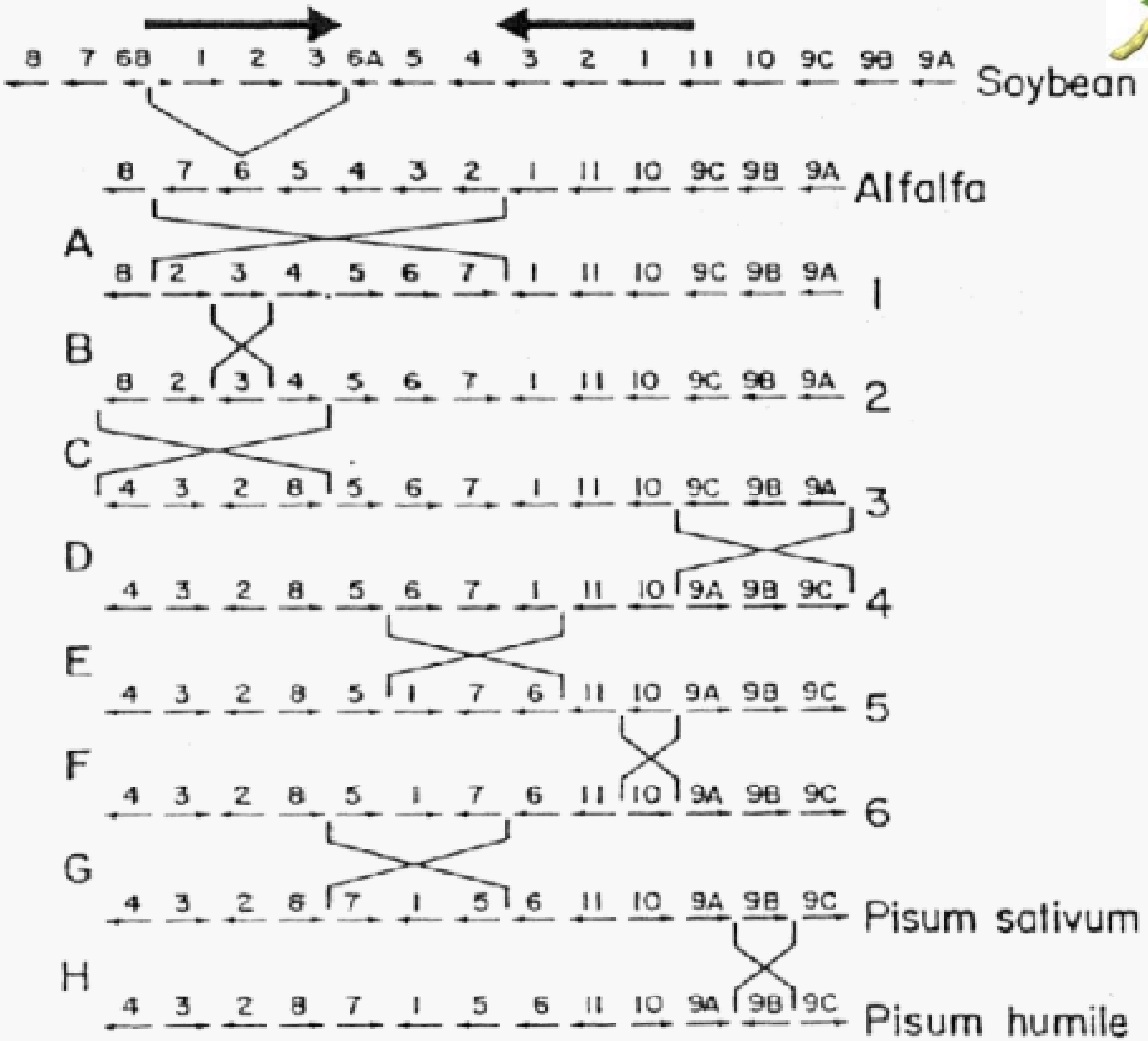
Goal: Find a shortest sequence of reversals that transform the given n-permutation to 1,2,...n

4	-3	1	-7	-6	-5	-2
4	-3	1	2	5	6	7
-4	-3	1	2	5	6	7
-2	-1	3	4	5	6	7
1	2	3	4	5	6	7

Reversal distance d: Length of shortest sequence

d=6


RADIATA PINE
A
E
C
D
F
A
psaA&B
psbC&D
16S
23S
rps12&7
psbA
atpA&F
atpH&I
rps2
rpoC
rpoB
psbE&F
petA
rbcL
atpB&E
psbB
petB&D
rpoA
rpl16
11.3
3.2
4.6
4.3
2.7
4.1
1.0
21.0
7.5
12.1
2.9
31.1
13.8
16S
23S
rps7&12
23S
16S
rps12&7
rpl23&2
psbA
atpA,F,H&I
rps2
rpoC&B
psbD&C
psaB&A
atpE&B
rbcL
petA
psbF&E
psbB
petB&D
rpoA
B
C
D
E
F
A
MUNG BEAN
PETUNIA



Palmer et. al, Current Genetics '88

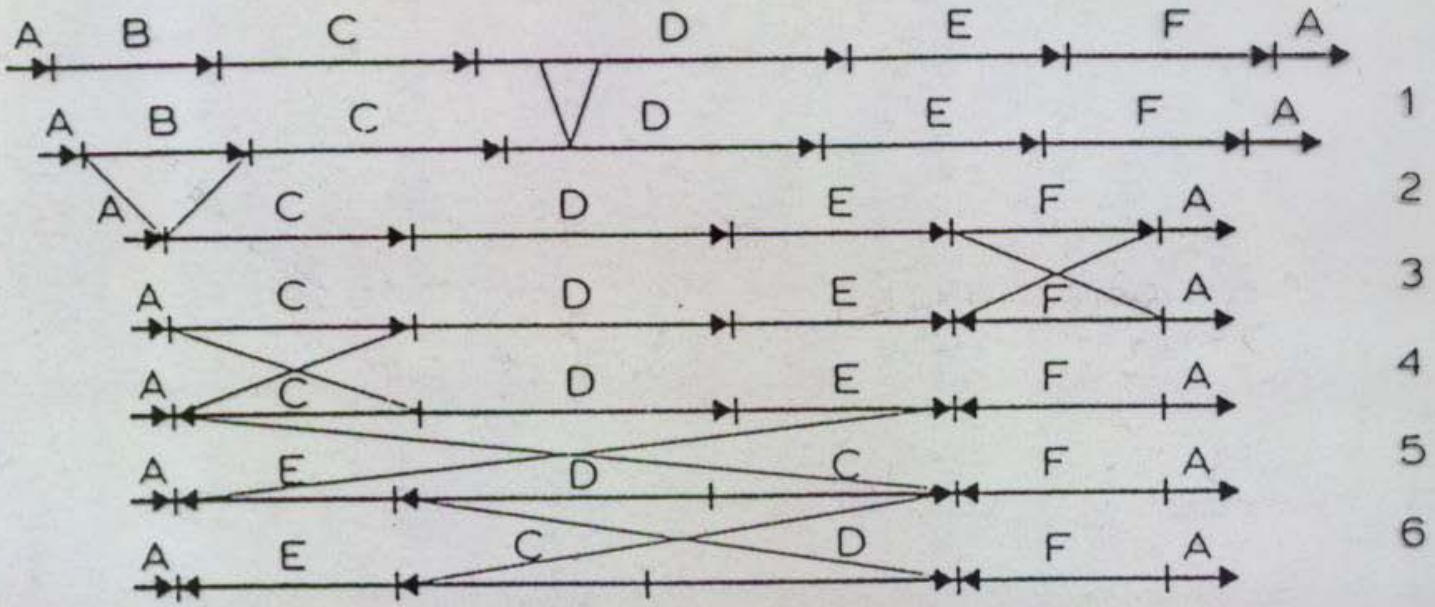
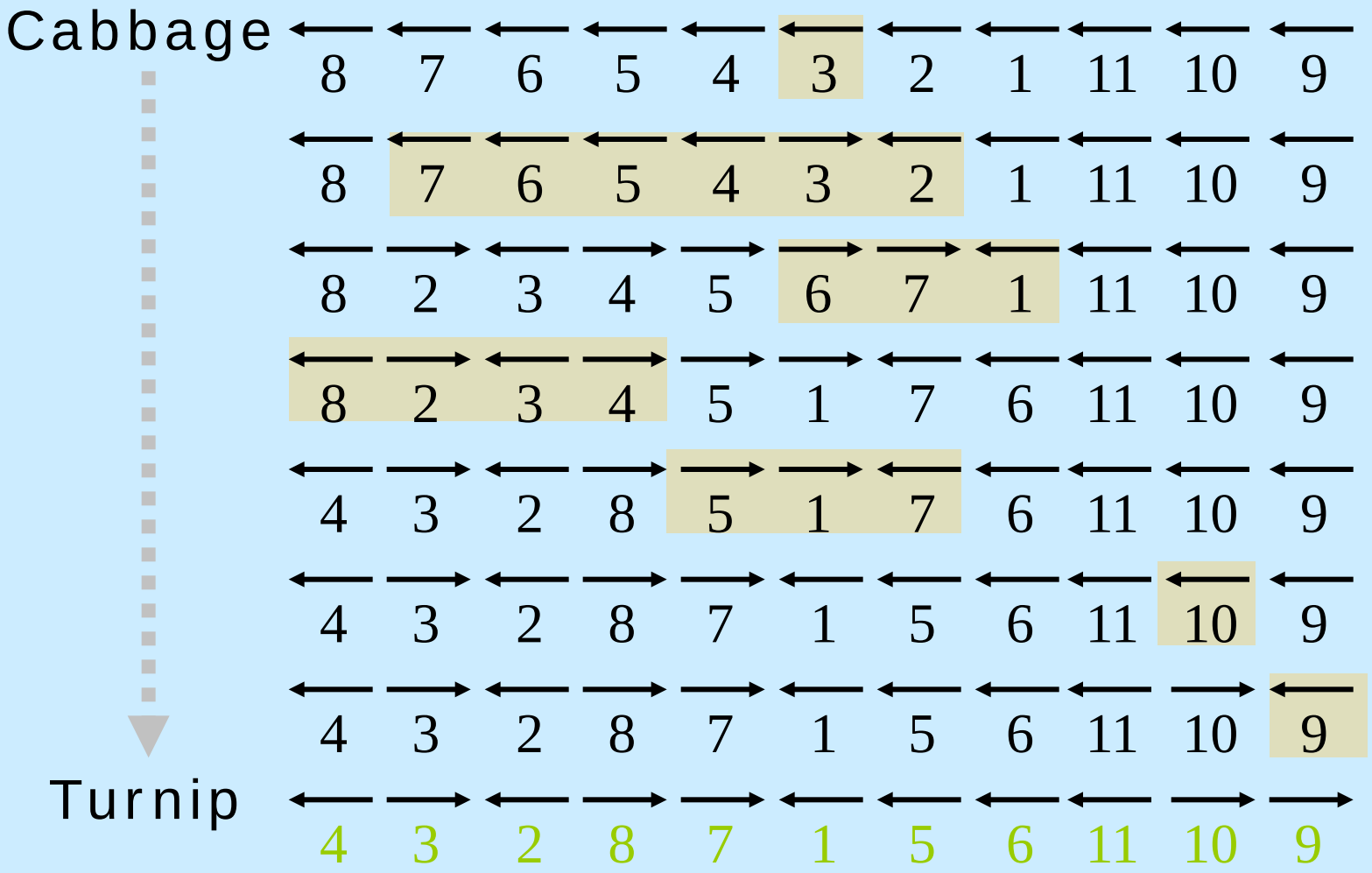


FIG. 4. Hypothesized deletions and inversions during evolution of the radiata pine chloroplast genome from a *Petunia*–mung bean-like ancestral genome shown in Fig. 3. Step 1 is deletion of a part of one repeat, similar to that seen in *Ginkgo* (12), the sole member of a different gymnosperm order. The evolutionary direction of the deletion is not clear (12), whereas the other five mutations shown are all clearly derived in a conifer-specific lineage. Step 2 is deletion of the inverted repeat. Steps 3–6 are inversions. The sequence of rearrangements that occurred during conifer evolution may differ from that presented.

Strauss et. al; PNAS '88

Sorting by Reversals



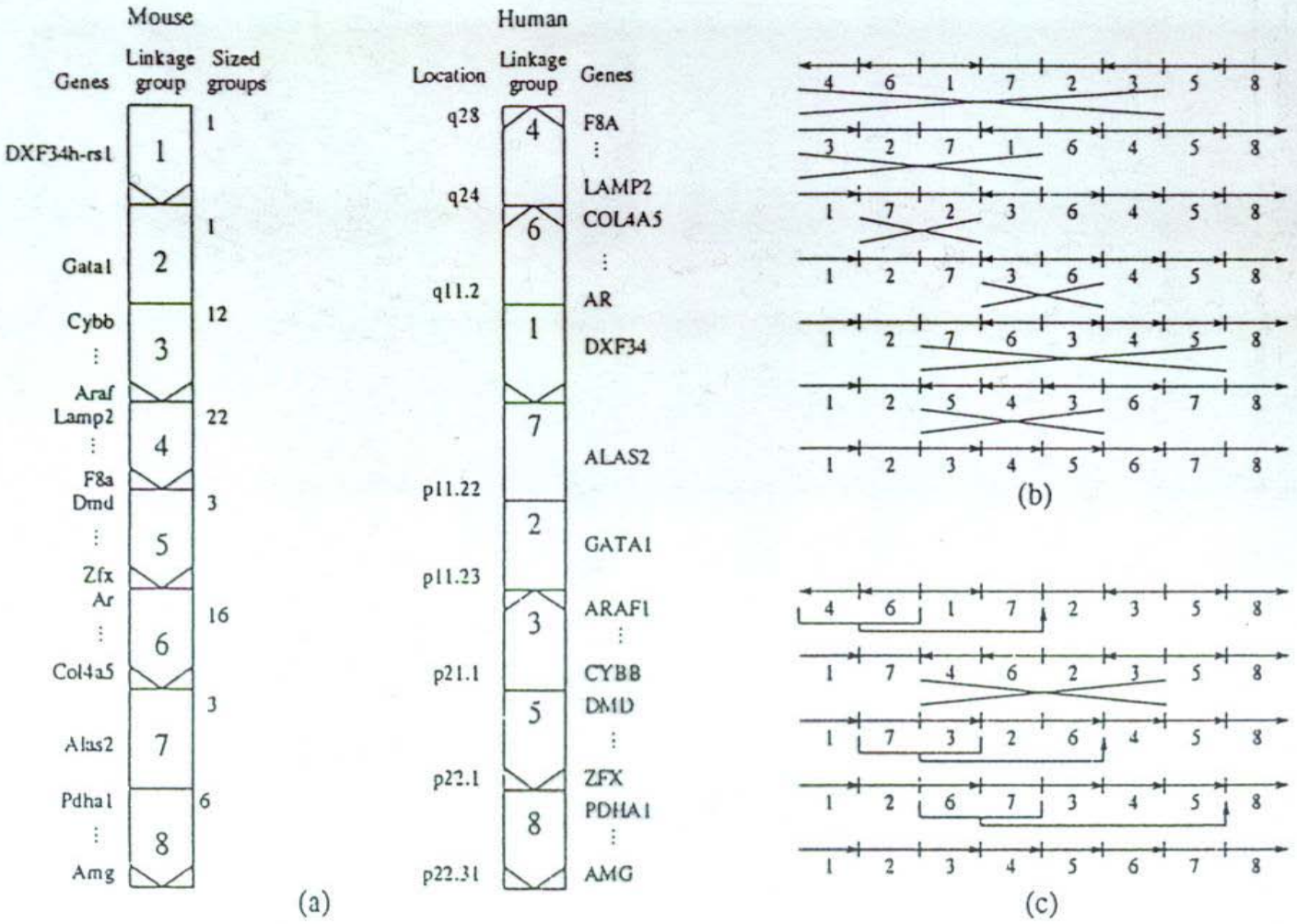


FIG. 4.—Transformation of human X chromosome into mouse X chromosome. *a*, Conserved linkage groups betwee X chromosomes. *b*, A most parsimonious evolutionary scenario for the transformation of human into mouse chromosor X chromosome evolves solely by inversions. *c*, A rearrangement scenario involving both inversions and transpositions.

Group Theoretic Viewpoint

Symmetric group of permutations S_n

Reversals form a generator set of S_n

Q: Given $\pi_1, \pi_2 \in S_n$, generators $g_1,\ldots,g_k$ find their *distance*: shortest product of generators that transforms π_1 to π_2

Even – Goldreich (81): NP-hard

Jerrum (85): PSPACE-complete

diameter: longest distance between two permutations

Q2: For generators $g_1,\ldots,g_k$ what is the diameter of S_n?

An aside: The Pancake Flipping Problem

- 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: Starting from the bottom of the stack, 2 prefix reversals at most to place a pancake in its right position → *2n – 2* steps total

Gates & Papadimitriou (79): Alg for sorting by *5/3 (n + 1) prefix reversals*

Discrete Mathematics 27 (1979) 47–57.

BOUNDS FOR SORTING BY PREFIX REVERSAL

William H. GATES
Microsoft, Albuquerque, New Mexico

Christos H. PAPADIMITRIOU*†
Department of Electrical Engineering, University of California, Berkeley, CA 94720, U.S.A.

Received 18 January 1978
Revised 28 August 1978

For a permutation σ of the integers from 1 to n, let $f(\sigma)$ be the smallest number of prefix reversals that will transform σ to the identity permutation, and let $f(n)$ be the largest such $f(\sigma)$ for all σ in (the symmetric group) S_n. We show that $f(n) \leq (5n+5)/3$, and that $f(n) \geq 17n/16$ for n a multiple of 16. If, furthermore, each integer is required to participate in an even number of reversed prefixes, the corresponding function $g(n)$ is shown to obey $3n/2 - 1 \leq g(n) \leq 2n+3$.

1. Introduction

We introduce our problem by the following quotation from [1]

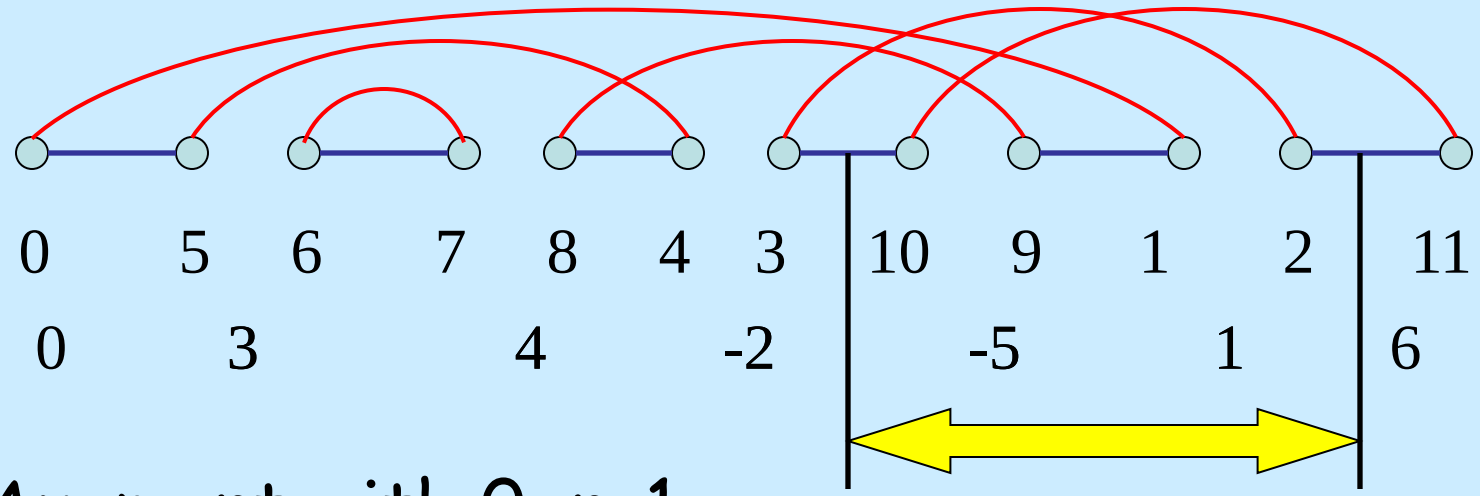
> The chef in our place is sloppy, and when he prepares a stack of pancakes they come out all different sizes. Therefore, when I deliver them to a customer, on the way to the table I rearrange them (so that the smallest winds up on top, and so on, down to the largest at the bottom) by grabbing several from the top and flipping them over, repeating this (varying the number I flip) as many times as necessary. If there are n pancakes, what is the maximum number of flips (as a function $f(n)$ of n) that I will ever have to use to rearrange them?

In this paper we derive upper and lower bounds for $f(n)$. Certain bounds were already known. For example, consider any stack of pancakes. An *adjacency* in this stack is a pair of pancakes that are adjacent in the stack, and such that no other pancake has size intermediate between the two. If the largest pancake is on the bottom, this also counts as one extra adjacency. Now, for $n \geq 4$ there are stacks of n pancakes that have no adjacencies whatsoever. On the other hand, a sorted stack must have all n adjacencies and each move (flip) can create at most one adjacency. Consequently, for $n \geq 4$, $f(n) \geq n$. By elaborating on this argument, M.R. Garey, D.S. Johnson and S. Lin [2] showed that $f(n) \geq n+1$ for $n \geq 6$.

For upper bounds—algorithms, that is—it was known that $f(n) \leq 2n$. This can be seen as follows. Given any stack we may start by bringing the largest pancake on top and then flip the whole stack: the largest pancake is now at the bottom,

* Research supported by NSF Grant MCS 77–01193.
† Current address: Laboratory for Computer Science, Massachusets, Institute of Technology, Cambridge, Ma 02139, USA.

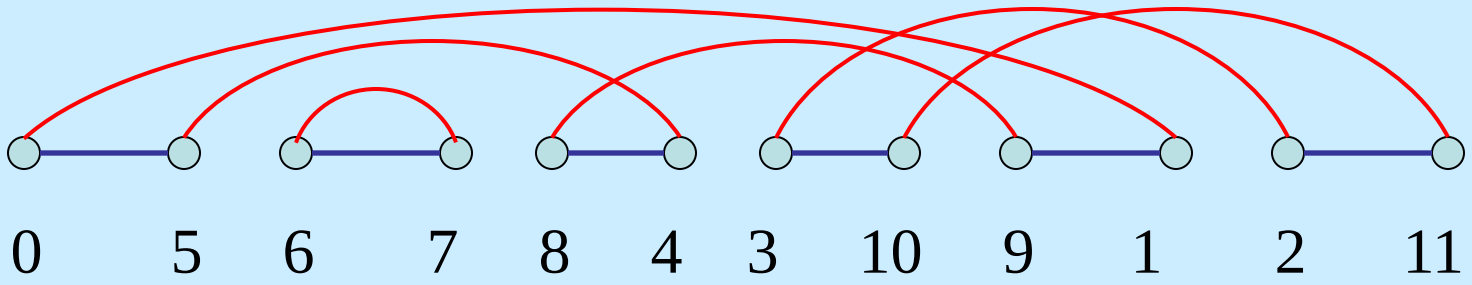
Back to SBR: The Breakpoint Graph



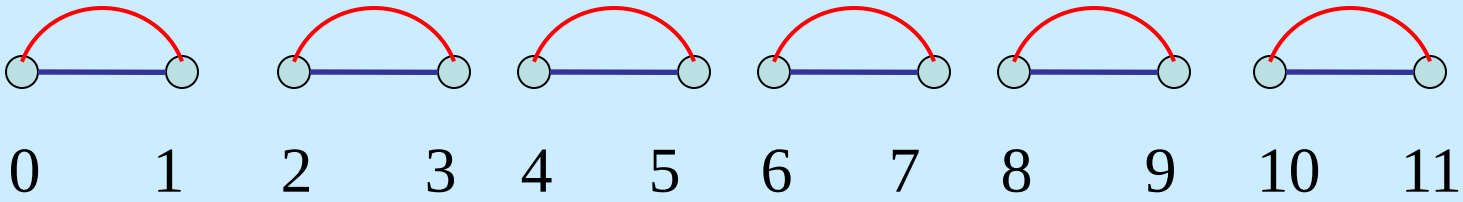
- Augment with 0, n+1
- Vertices 2i-1, 2i for +i, 2i, 2i-1 for -i
- Blue edges between adjacent vertices π_{2i} π_{2i+1}
- Red edges between consecutive labels 2i,2i+1

- Allow only reversals that cut after even positions

GOAL: Sort a given breakpoint graph



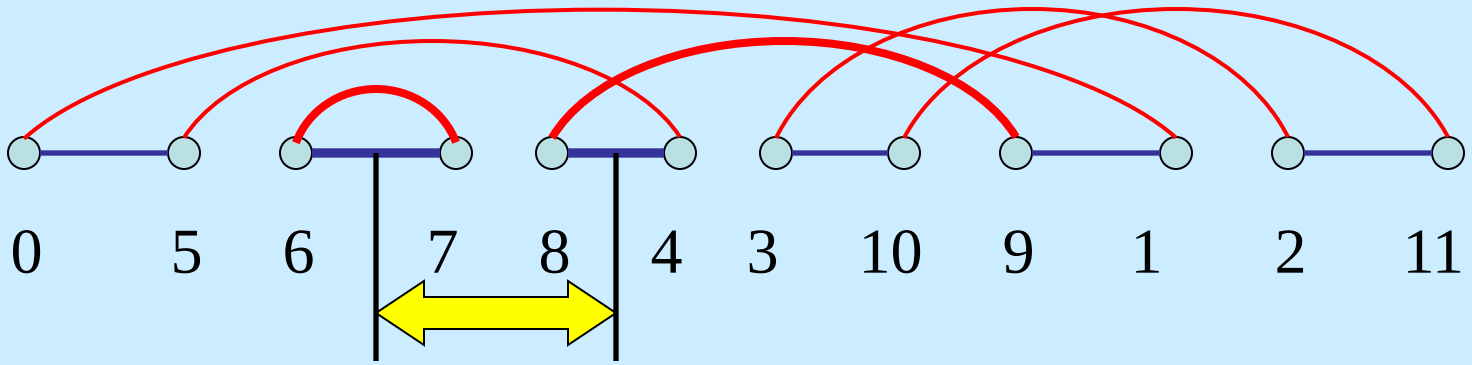
into n+1 trivial cycles



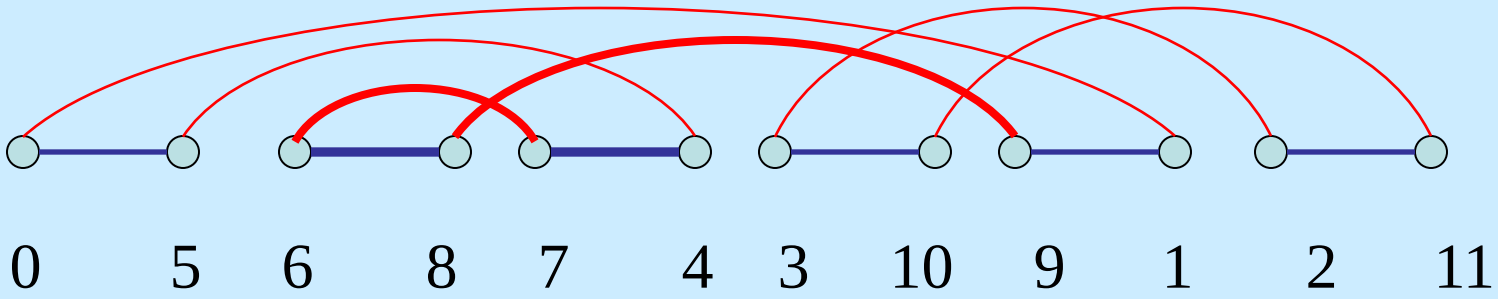
⇒ Try to increase number of cycles at each step

The impact of a reversal

Def: A reversal *acts* on two blue edges

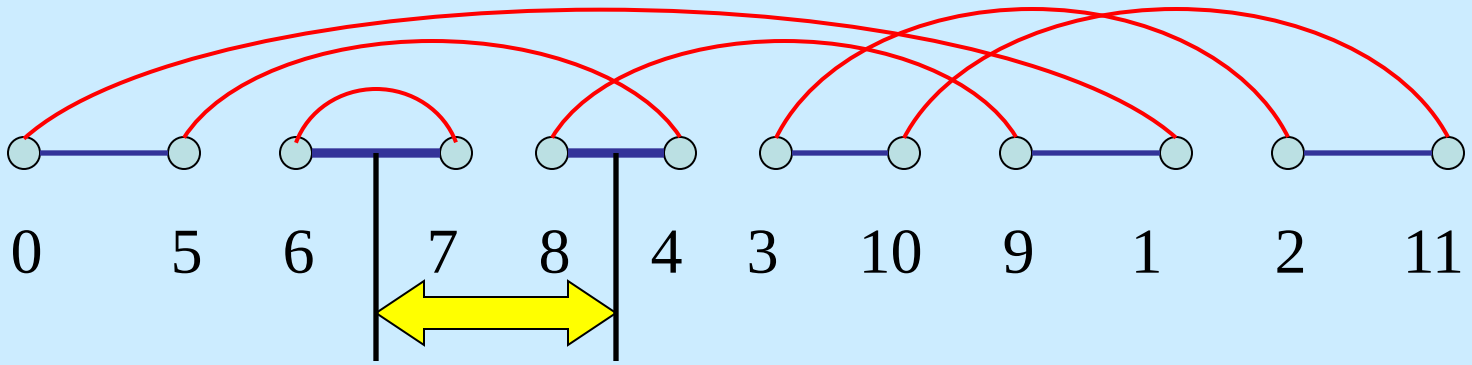


cutting them and re-connecting them

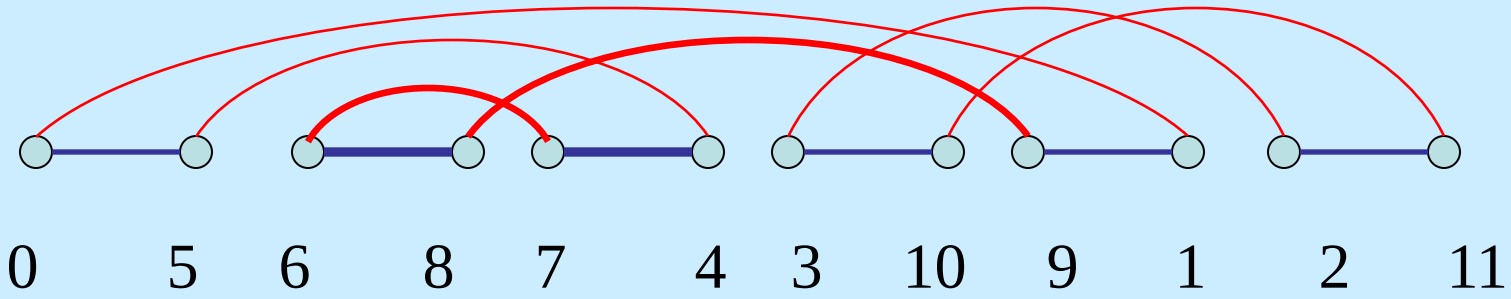


The impact of a reversal (2)

A reversal can either…

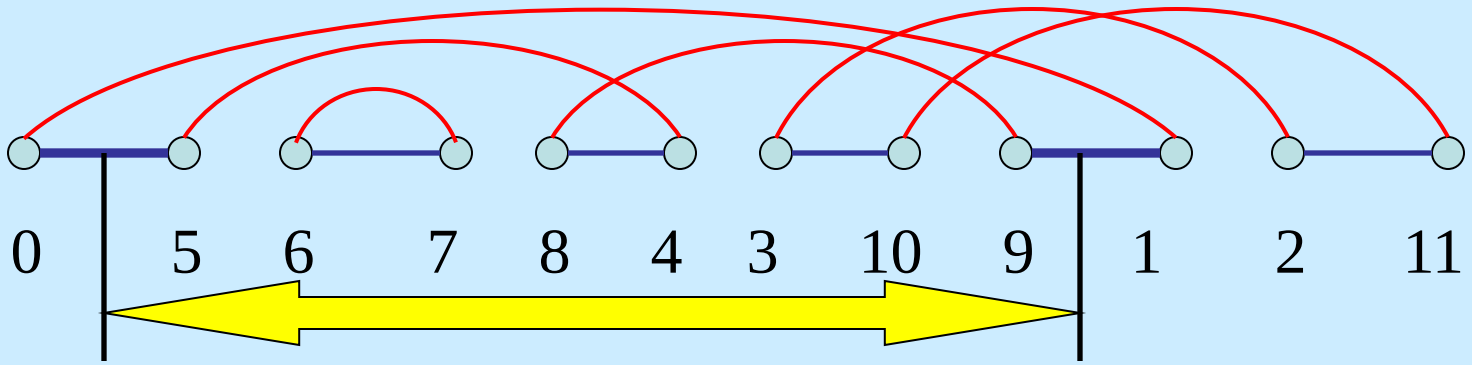


Act on two cycles, joining them (bad!!)

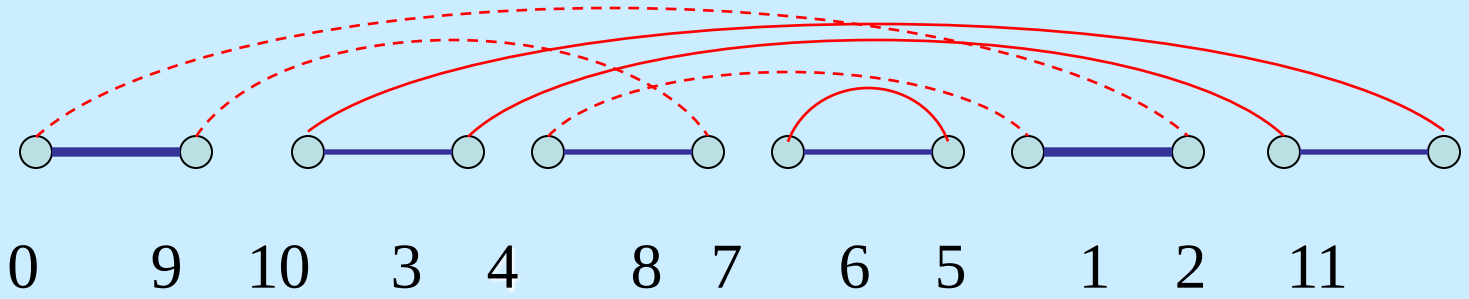


The impact of a reversal (3)

… or:

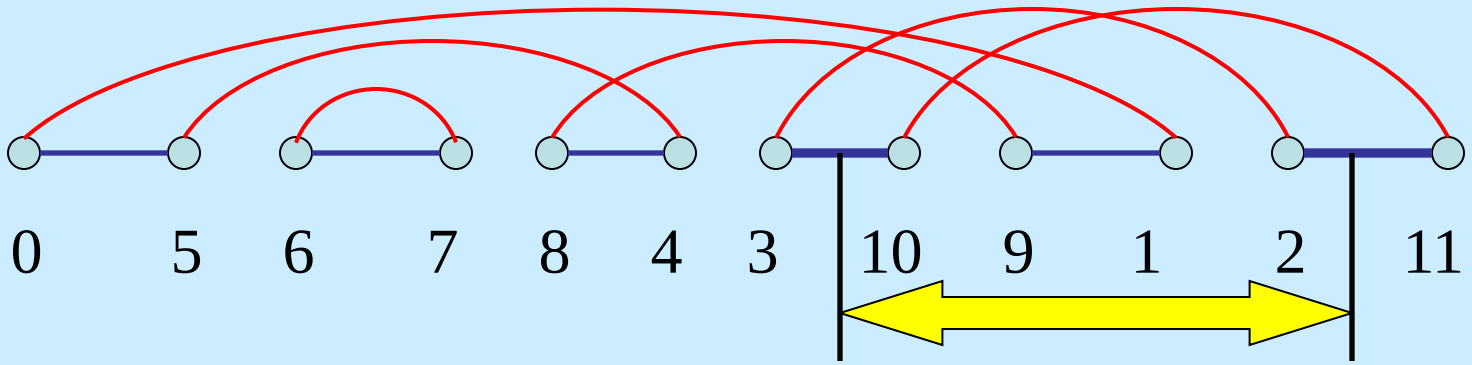


Act on one cycle, changing it (profitless)

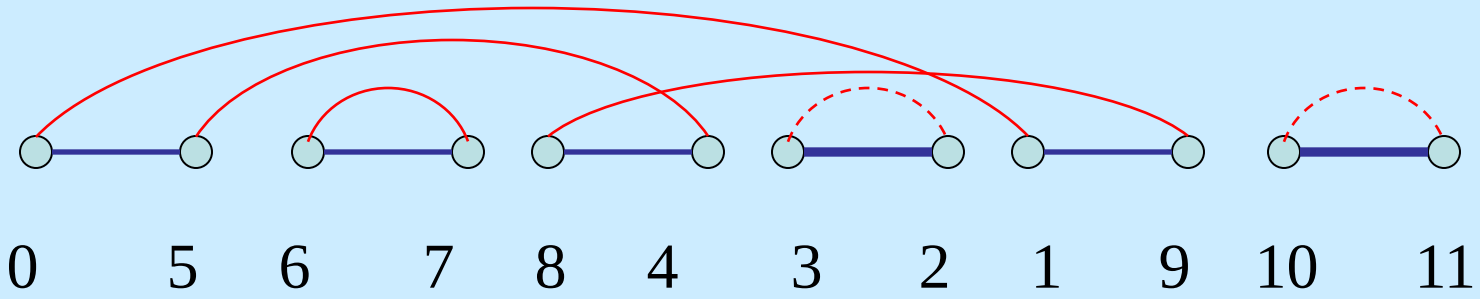


The impact of a reversal (4)

… or:



Act on one cycle, splitting it (***good*** **reversal**)



Basic Theorem (Bafna, Pevzner 93)

$$d(\pi) \geq n + 1 - c(\pi)$$

where *d* = *reversal distance*,

c = # cycles.

Proof: Every reversal changes *c* by at most 1.

Hannenhalli & Pevzner Theory (95)

Thm: $d(\pi) = n + 1 - c(\pi) + h(\pi) + f(\pi),\ f(\pi) \in \{0,1\}$

h – "hurdles" a parameter for reflecting interrelations of difficult cycles
f – "fortress" an additional parameter for a particular combination of hurdles. Can be 0 or 1

HP95 constructive proof;
Implies an $O(n^4)$ algorithm for SBR
Many improvements since.

Sorting by Signed Reversals: History

Sankoff (90,92)

Kececioglou – Sankoff (95) 2-approximation

Bafna – Pevzner (94) 1.5-approximation

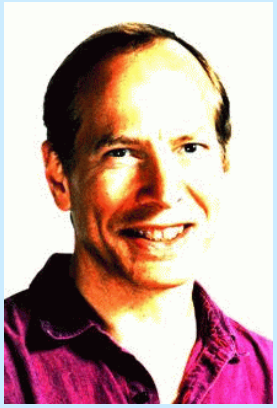
Rich combinatorial structure (KS95, KR95, BP95, H95,…)

- Hannenhalli – Pevzner (95) first poly alg $O(n^4)$

Caprara (96) unsigned problem is NP-hard

- Berman – Hannenhalli (96) O(n2a(n)) implementation

- Kaplan Shamir Tarjan (99) $O(n^2)$ alg, based on HP95, much simpler



Sorting by Signed Reversals: History (2)

- ♣ Bergeron (01,03) – simplified theory, $O(n^3)$

Bader, Moret, Yan (01) O(n) alg for reversal distance

- Bergeron (03) simple presentation, $O(n^3)$
- Ozery-Flato & Shamir (03) $\Omega(n^3)$ for Bergeron's alg
- Verbin & Kaplan (03) efficient data structure for reversals
- Tannier, Bergeron, Sagot (04) $O(n^{1.5} (\log n)^{0.5})$
- Swenson Rajan Lin Moret (09) O(n log n)





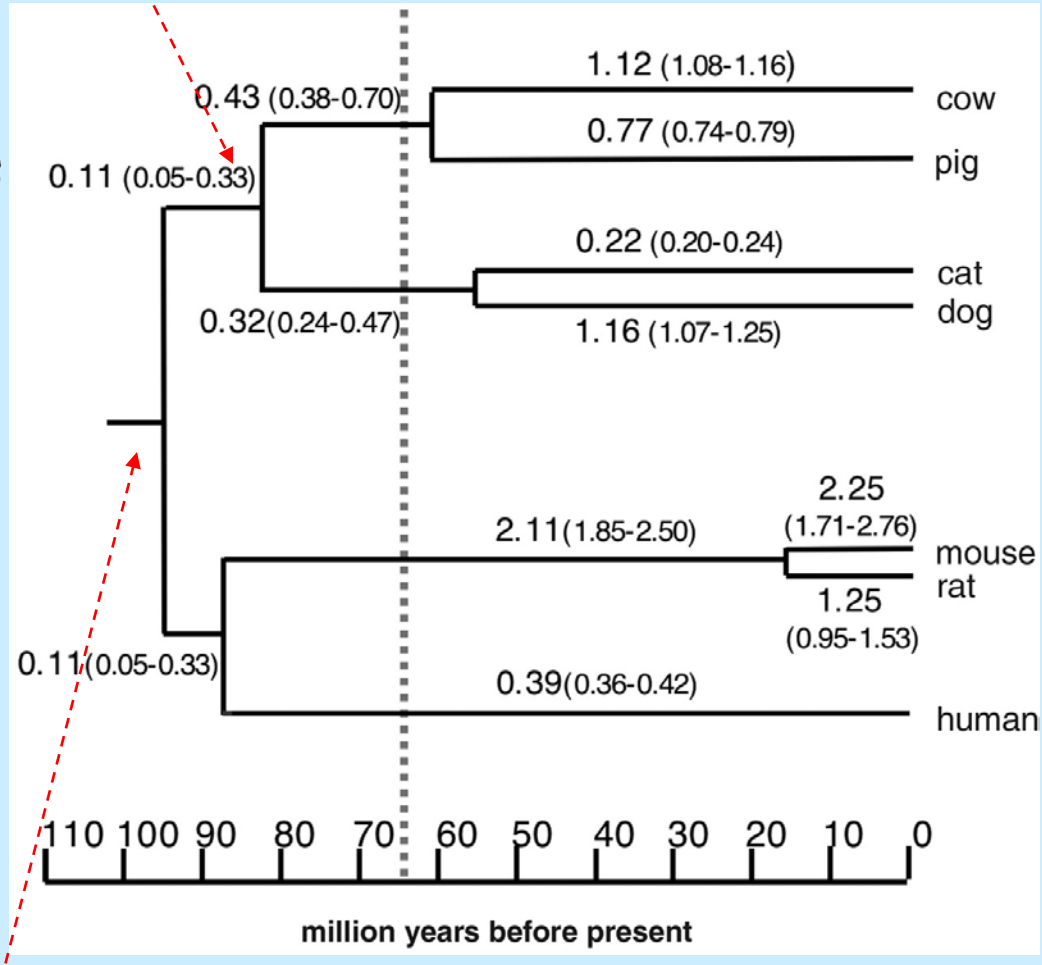

More on Genome Rearrangements

- Hannenhalli, Pevzner 95: Poly alg. for sorting by reversals, translocations, fusions and fissions
- Reconstructed the Human-mouse evolution scenario with 131 events
- Multi species GR phylogenies
- Hot debate on breakpoint reuse
- …

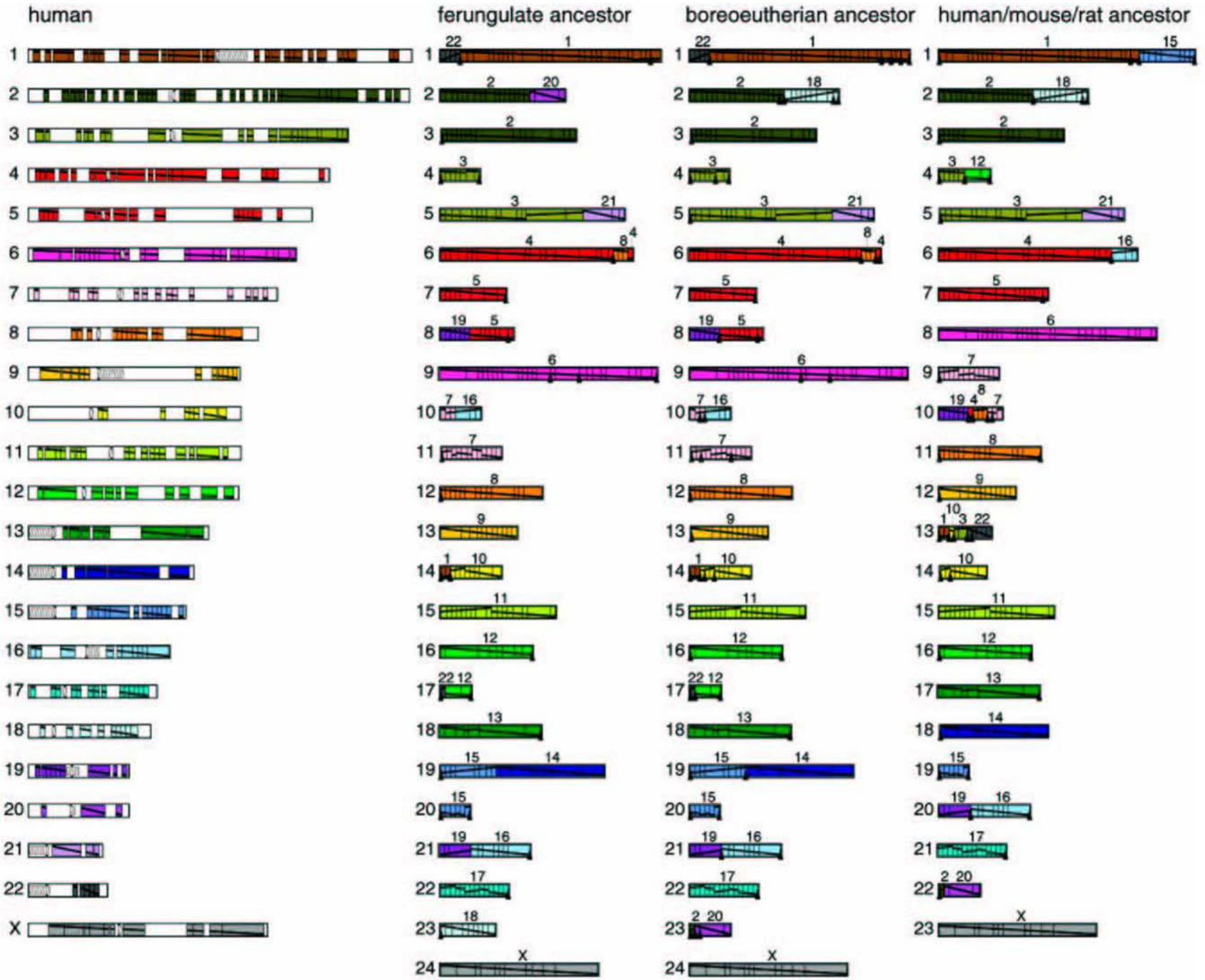
Murphy et al Science 2005

Ferungulate ancestor

Fig. 3. Rates of chromosome breakage during mammalian evolution. The time scale is based on molecular divergence estimates (*19*). Rates (above the branches, in breaks per million years and 95% confidence intervals) were calculated using the total number of lineage, order, or superordinal breakpoints defined by the multispecies breakpoint analysis, and dividing these by the estimated time on the branch of the tree. The vertical gray dashed line indicates the K-T boundary, marking the abrupt extinction of the dinosaurs at 65 Ma and preceding the appearance of most crown-group placental mammal orders in the Cenozoic Era (*19*).



Boreoeutherian (placental mammals) ancestor


human
ferungulate ancestor
boreoeutherian ancestor
human/mouse/rat ancestor

- **Fig. 2.** Genome architecture of the ancestors of three mammalian lineages computed by MGR (*33*) from the seven starting genomes and compared to the human genome (far left). Each human chromosome is assigned a unique color and is divided into blocks corresponding to the seven-way HSBs common to all species. The size of each block is approximately proportional to the actual size of the block in human. Physical gaps between blocks are shown in human to give an indication of the coverage. Also in human, the heterochromatic/centromere regions are denoted by hatched gray boxes. Numbers above the reconstructed ancestral chromosomes indicate the human chromosome homolog. Diagonal lines within each block (from top left to bottom right) indicate the relative order and orientation of genes within the block. Black arrowheads under the ancestral chromosomes indicate that the two adjacent HSBs separated by the arrowhead were not found in every one of the most parsimonious solutions explored; these are considered "weak" adjacencies. Arrowheads at the ends of HSB chromosomes indicate that some alternative solutions placed these chromosome-end HSBs adjacent to HSBs from other chromosomes. [View Larger Version of this Image (46K GIF file)]

Sorting genomes by DCJ operations

Bergeron, Mixtacki, Stoye. A unifying view of Genome Rearrangements. WABI 2006.

Slides based in part on Ghada Badr

http://www.site.uottawa.ca/~turcotte/teaching/csi-5126/lectures/09/1/GenomeRearrangement_PartII_Ghada.ppt

Rearrangement Problems

Our problem:

Given two **genomes** and a set of possible evolutionary **events** (operations), find a **shortest** sequence of events transforming those genomes into one another.

Two classical problems

- Computing the **distance** d(π).
- Computing one optimal **sorting sequence** of events.

Rearrangement Operations

Can we have a **unifying** framework in which circular and linear chromosomes can coexist throughout evolving genomes?

Can we have a unifying view of Genome Rearrangements? (Bergeron 2006)

A Double Cut and Join Operation DCJ was introduced.

Rearrangement Operations -DCJ

- **Double Cut-and-Join** DCJ was first proposed by *Yancopoulos et. al.* (2005).
- Allows to model many **classical** operations (inversions, translocations, fissions, fusions) with a **single** operation. Others (transposition, block interchanges) in two.
- Model assumes the coexistence of both **linear** and **circular** chromosomes. There is some evidence for this in genomes.
- Both the DCJ **sorting** and **distance** problems can be solved in O(n) time by *Bergeron et. al.* (2006)

Adjacencies and telomeres

- A **“gene” a** is an oriented sequence of DNA that starts with a *tail* **at** and ends with a *head* **ah**.

- Two consecutive genes do not necessarily have the same orientation, thus **adjacency** of two consecutive genes **a** and **b**, can be of four different types:

  [ah,bt],[ah,bh],[at,bt],[at,bh]

  ➔➔ , ➔➔ , ➔➔ , ➔➔

  (we use [] and not {} for sets to avoid a PPT bug…)

- An extremity that is not adjacent to any other gene is called **telomere.** It is denote by a **singleton** set: [ah] or [at].

- We can use adjacencies to represent both genomes with **multiple** or **uni**-chromosomes.

Genome representation

- A **genome** is a set of adjacencies and telomeres such that the tail or head of any gene appears in exactly **one** adjacency or telomere.

Example

Genome A: chr1: a c -d
chr2: b e
chr3: f g

➔

Replace each gene by two extremities

at ah ct ch dh dt
bt bh et eh
ft fh gt gh

Adjacencies :[ah, ct][ch, dh][bh, et] [fh, gt]

Telomere:[at] [dt] [bt] [eh][ft][gh]

Note 2: if a genome has N genes, a adjacencies , t telomeres, then N=a+t/2

A = [[at][ah, bt][bh, ct][ch, dt][dh] [et] [eh,ft] [fh,gt] [gh]]

Note: a chromosome is identical to its inverted copy

Double cut and join (DCJ) - definition

Definition 1. *The* double cut and join *(DCJ) operation acts on two vertices u and v of a graph with vertices of degree one or two in one of the following three ways:*

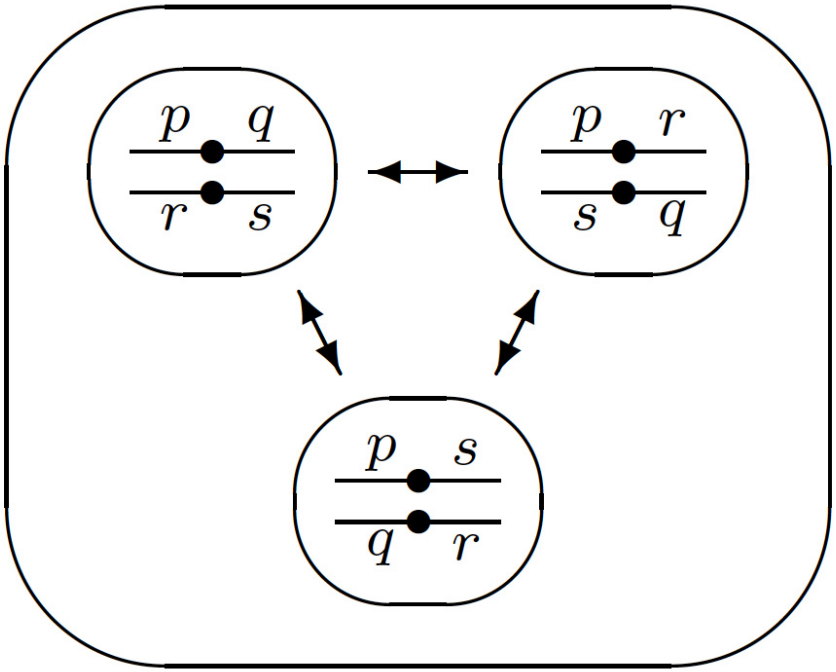
(a) If both $u = \{p, q\}$ and $v = \{r, s\}$ are internal vertices, these are replaced by the two vertices $\{p, r\}$ and $\{s, q\}$ or by the two vertices $\{p, s\}$ and $\{q, r\}$.
(b) If $u = \{p, q\}$ is internal and $v = \{r\}$ is external, these are replaced by $\{p, r\}$ and $\{q\}$ or by $\{q, r\}$ and $\{p\}$.
(c) If both $u = \{q\}$ and $v = \{r\}$ are external, these are replaced by $\{q, r\}$.

In addition, as an inverse of case (c), a single internal vertex $\{q, r\}$ can be replaced by two external vertices $\{q\}$ and $\{r\}$.

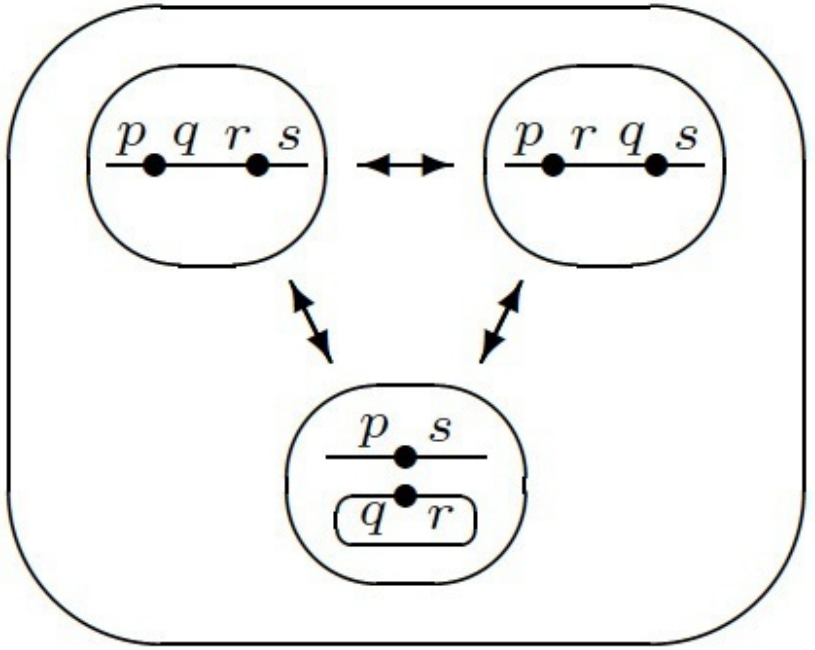
Rearrangement Operations -DCJ

- **DCJ operations**:

a) [p,q][r,s] ⟶ [p,r][s,q] or [p,s] [q,r]



Translocation



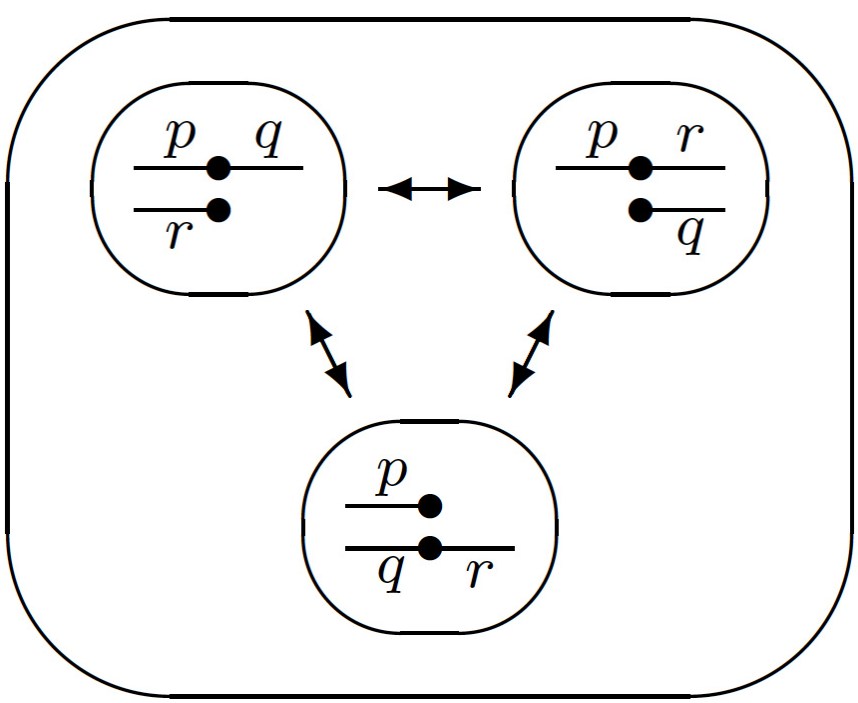
Inversion

Excision (splicing out a cycle)

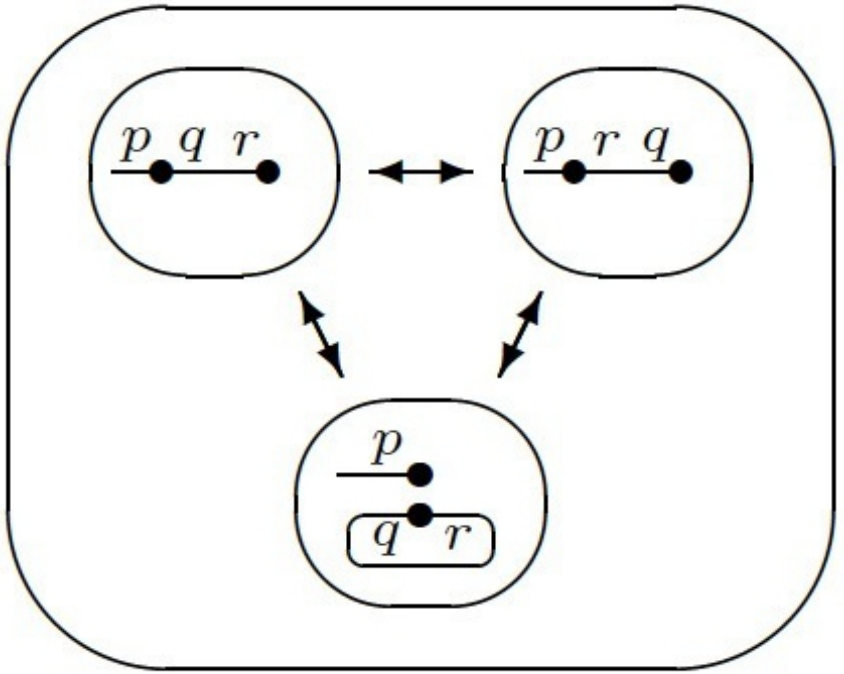
Rearrangement Operations - DCJ

- **DCJ operations**:

b) [p,q][r] ⟶ [p,r][q] or [p][q,r]



Unbalanced (tail) translocation



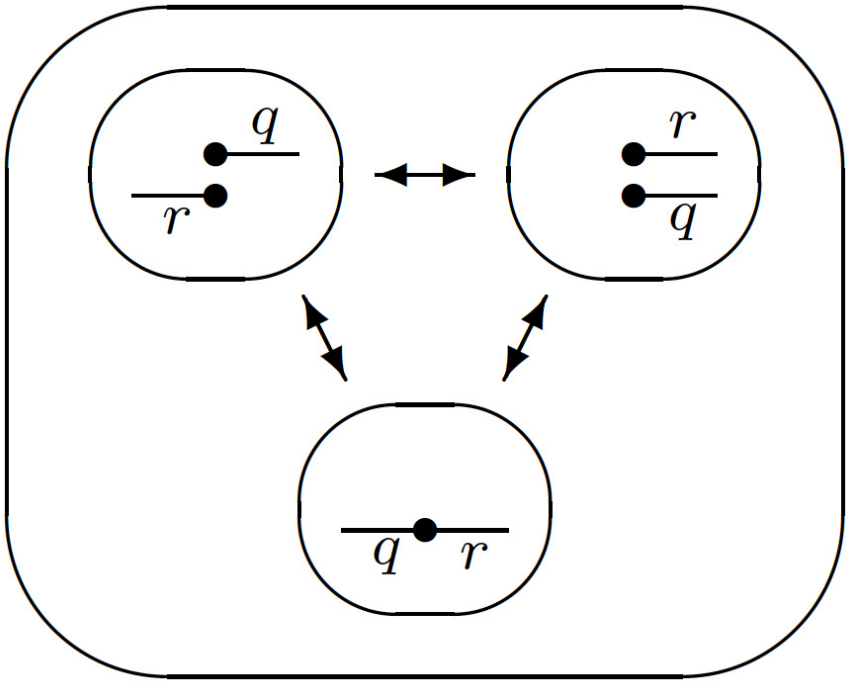
Inversion

Excision (splicing out a cycle)

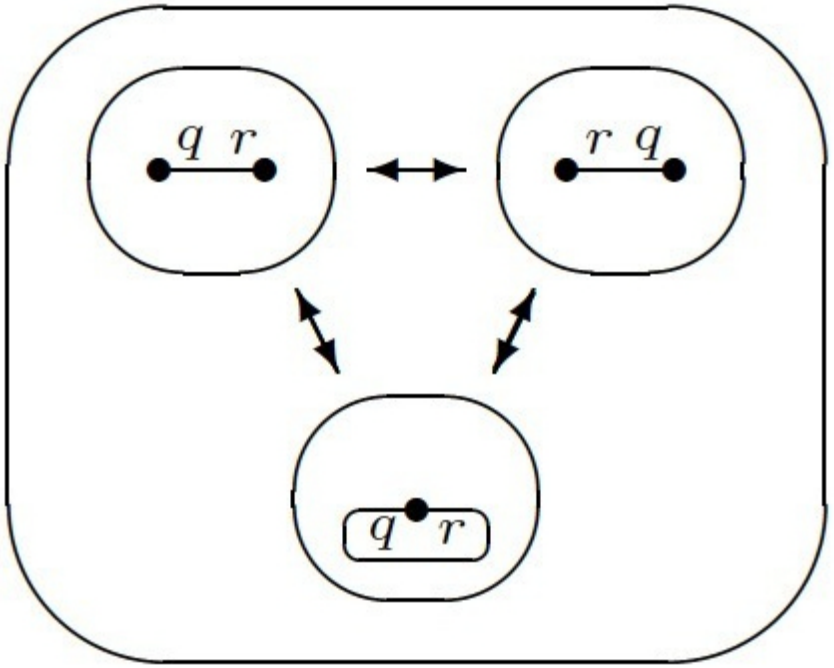
Rearrangement Operations -DCJ

- **DCJ operations**:

c) [q] [r] ⟷ [q,r]



Fusion/fission



Circularization/linearization

Lemma 1: A DCJ operation changes the number of linear or circular components by ≤ 1

Pf: case analysis

(Q: which case did we not consider?)

DCJ Example

Genome A: chr1: a | c -d
chr2: b e
chr3: f | g

Adjacencies and telomeres:

[ah, | ct][ch, dh] [bh, et] [fh, | gt] [at]]dt] [bt] [eh][ft][gh]

[ah,ct][fh, gt] → [ah,fh][ct,gt] ➔ Genome A: chr1: a -f
chr2: b e
chr3: d -c g

[ah,ct][fh, gt] → [ah,gt][ct,fh] ➔ Genome A: chr1: a g
chr2: b e
chr3: f c -d

DCJ sorting and Distance problems

Problem: Given two genomes A and B defined on the **same** set of genes, find a **shortest** sequence of DCJ operations that transforms A into B. The **length** of such a sequence is called the DCJ distance between A and B, dcj(A,B).

DCJ sorting and Distance problems

Example:

Replace each gene by two extremities

Genome A: chr1: a c -d
chr2: b e
chr3: f g

➔ at ah ct ch dh dt
bt bh et eh
ft fh gt gh

Genome B: chr 1: a b c d ➔ at ah bt bh ct ch dt dh
chr 2: e f g et eh ft fh gt gh

Get adjacencies and telomeres for each genome:

A =[[ah, ct][ch, dh] [bh, et] [fh, gt] [at] [dt]]bt] [eh][ft][gh]]

B = [[at][ah, bt][bh, ct][ch, dt][dh] [et] [eh,ft] [fh,gt] [gh]]

Greedy Alg to sort by DCJ

[ah, ct][ch, dh] [bh, et] [fh, gt] [at] [dt]]bt] [eh][ft][gh]

Genome A: chr1: a c -d
chr2: b e
chr3: f g

[ah, bt][ch, dh] [bh, et] [fh, gt] [at] [dt]]ct] [eh][ft][gh]

Genome A: chr1: a b e
chr2: c -d
chr3: f g

[ah, bt] [ch, dh] [bh, ct] [fh, gt] [at] [dt]]et] [eh][ft][gh]

Genome A: chr1: a b c -d
chr2: e
chr3: f g

[ah, bt] [ch, dt] [bh, ct] [fh, gt] [at] [dh] [et]]eh] [ft][gh]

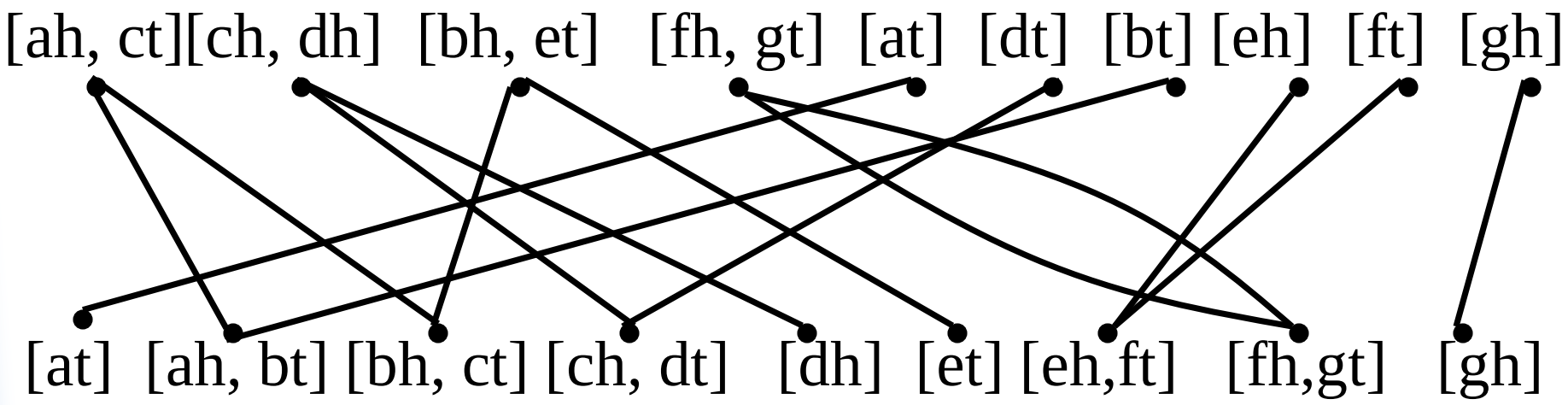
Genome A: chr1: a b c d
chr2: e
chr3: f g

[at][ah, bt][bh, ct][ch, dt][dh] [et] [eh,ft]]fh,gt] [gh]

Genome B: chr1: a b c d
chr2: e f g

The adjacency graph AG(A,B) of genomes A, B

A bipartite graph of the intersection of adj&tel in the two genomes:



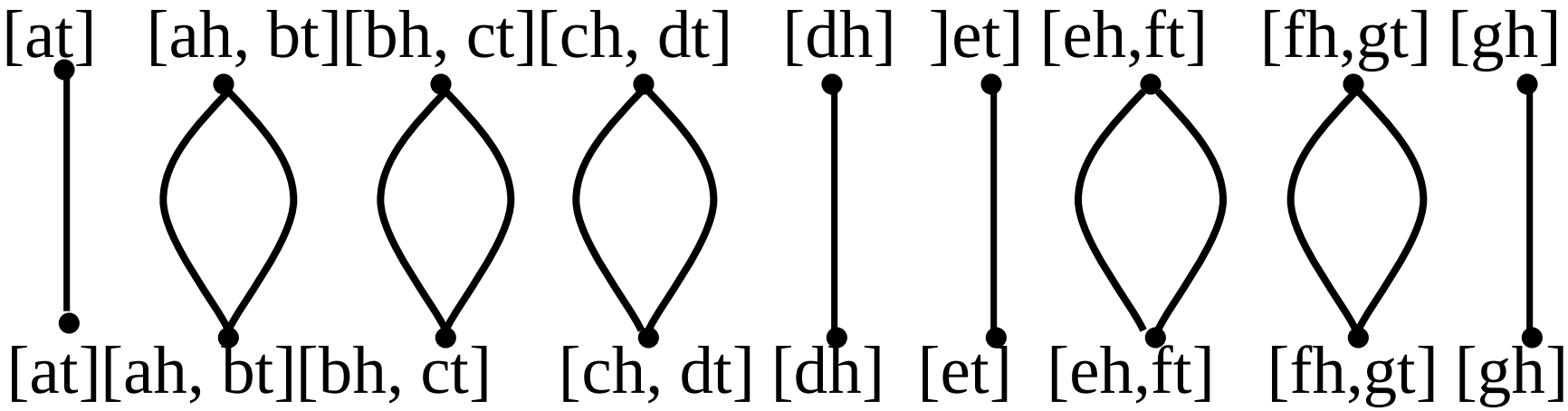
Vertices: adjacencies and telomeres

Edges: between vertices that have common elements.

A union of paths and cycles.

Graph can be easily constructed in O(n) time and space

The adjacency graph AG(A,A)

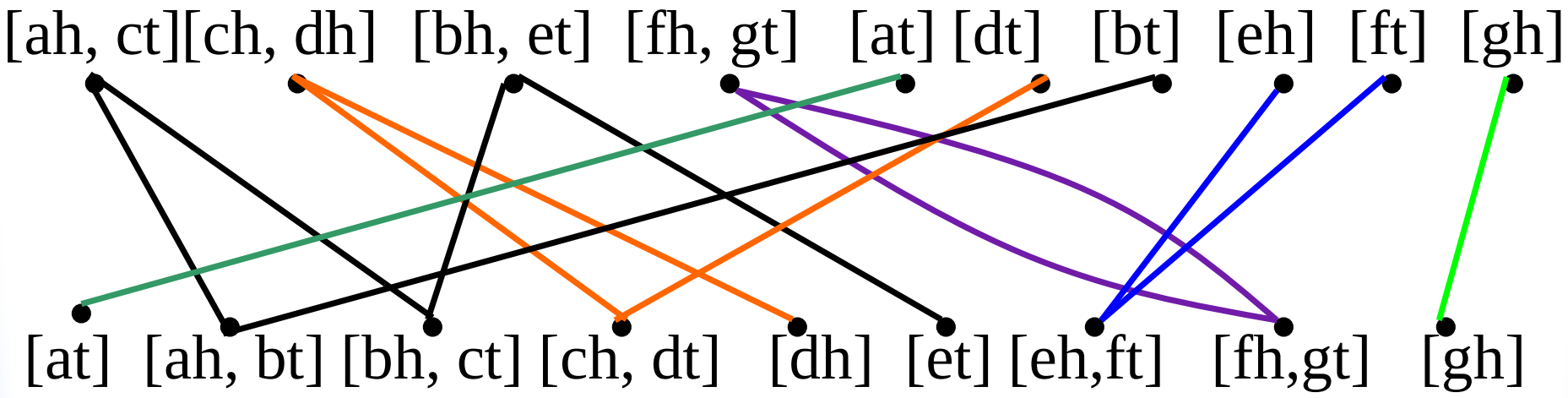


C: no. of cycles. I: no. of odd paths.

When sorted :N = C + I/2

DCJ sorting and Distance problems

Adjacency Graph (bipartite graph):



1cycle

4odd paths

1even path

Lemma 2: For A, B N-gene genomes A=B iff N=C+I/2

Pf: ← A=B with a adjacencies, t telomeres
→ a=C, t=I. N=a+t/2 = C+I/2

→ G adj. graph of A,B satisfies N=C+I/2.

A has a adjacencies, t telomeres → N=a+t/2

Each cycle has ≥1 adjacency → C≤a

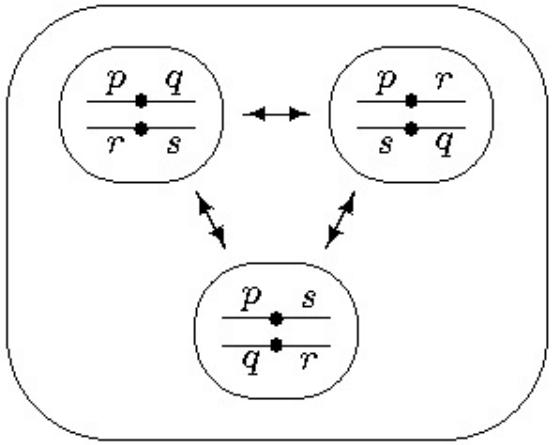
Each odd path has 1 telomere of A → t≤I

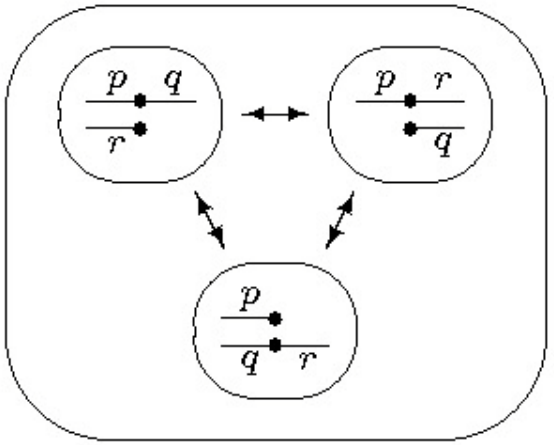
N=a+t/2=C+I/2 → a=C, I=t

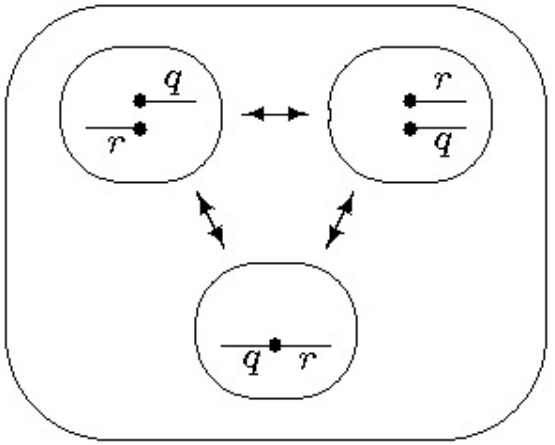
→ All cycles of length 2, all odd paths of length 1 → B=A

Lemma 3: A DCJ operation changes the number of odd paths by -2,0 or 2

Pf: simple case analysis. Some cases:







Lemma 4: For genomes A, B with the same set of N genes, $d_{DCG}(A,B) \geq N-(C+I/2)$

Pf: One DCJ operation may change the number of cycles or the number of odd paths – but not both.

Each operation changes C by ≤ 1 (Lemma 1)

Each operation changes I by ≤ 2 (Lemma 3)

→ Each operation changes C+I/2 by ≤ 1

When terminating N=C+I/2 (lemma 2)

→ $d_{DCG}(A,B) \geq N-(C+I/2)$

DCJ sorting algorithm

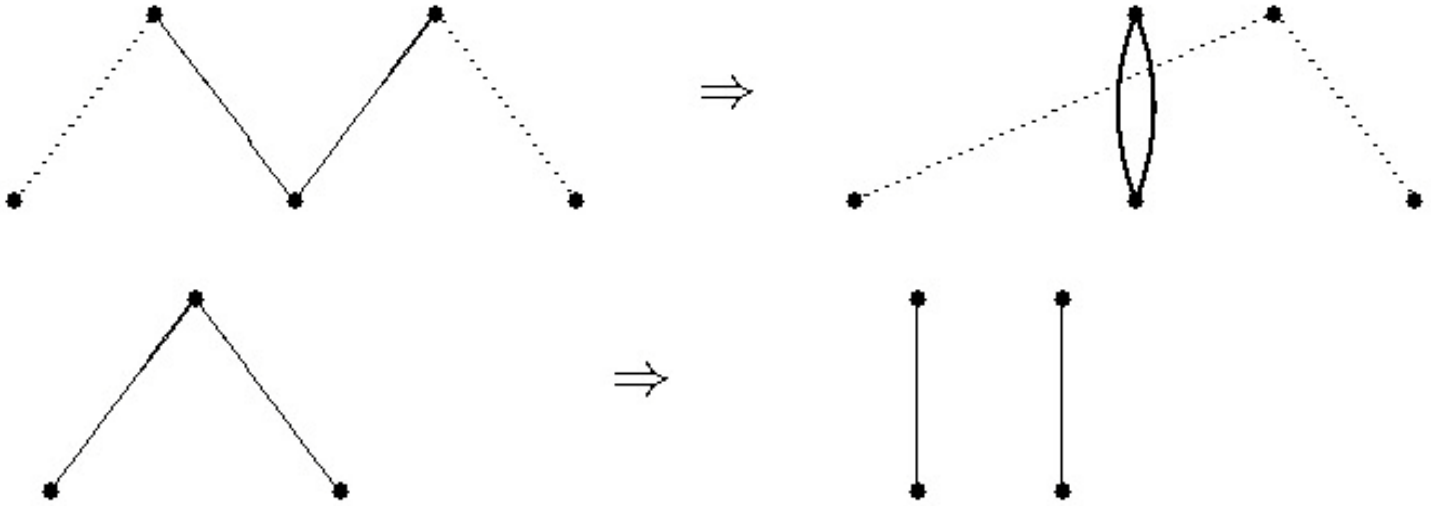
Algorithm 2 (Greedy sorting by DCJ)

```
 1: for each adjacency {p, q} in genome B do
 2:     let u be the element of genome A that contains p
 3:     let v be the element of genome A that contains q
 4:     if u ≠ v then
 5:         replace u and v in A by {p, q} and (u \ {p}) ∪ (v \ {q})
 6:     end if
 7: end for
 8: for each telomere {p} in genome B do
 9:     let u be the element of genome A that contains p
10:     if u is an adjacency then
11:         replace u in A by {p} and (u \ {p})
12:     end if
13: end for
```

O(n) time (ex)

Theorem: $d_{DCG}(A,B)=N-(C+I/2)$ and the greedy alg is optimal

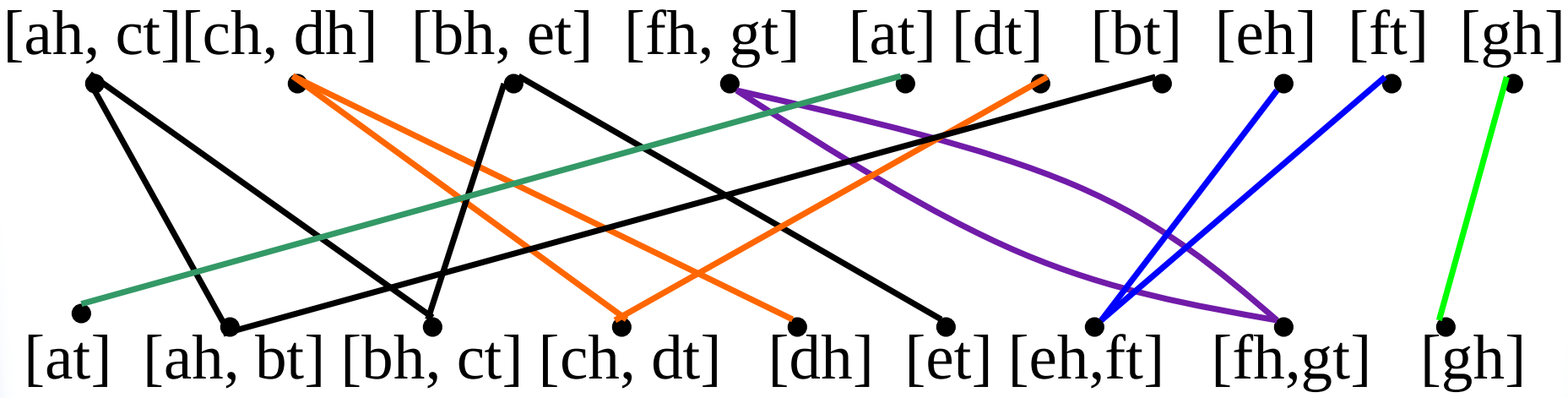
Pf. Effect of an iteration:



Each iteration increases C by 1 or I by 2, so Lemma 4 implies the equality and the optimality.

DCJ sorting and Distance problems

Adjacency Graph (bipartite graph):



1cycle

4odd paths

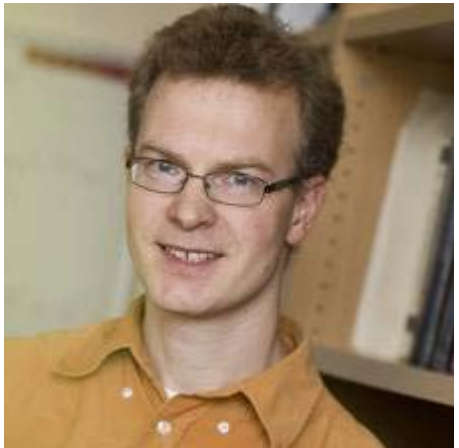
1even path

dcj(A,B) = n - (cycles + oddPath/2)

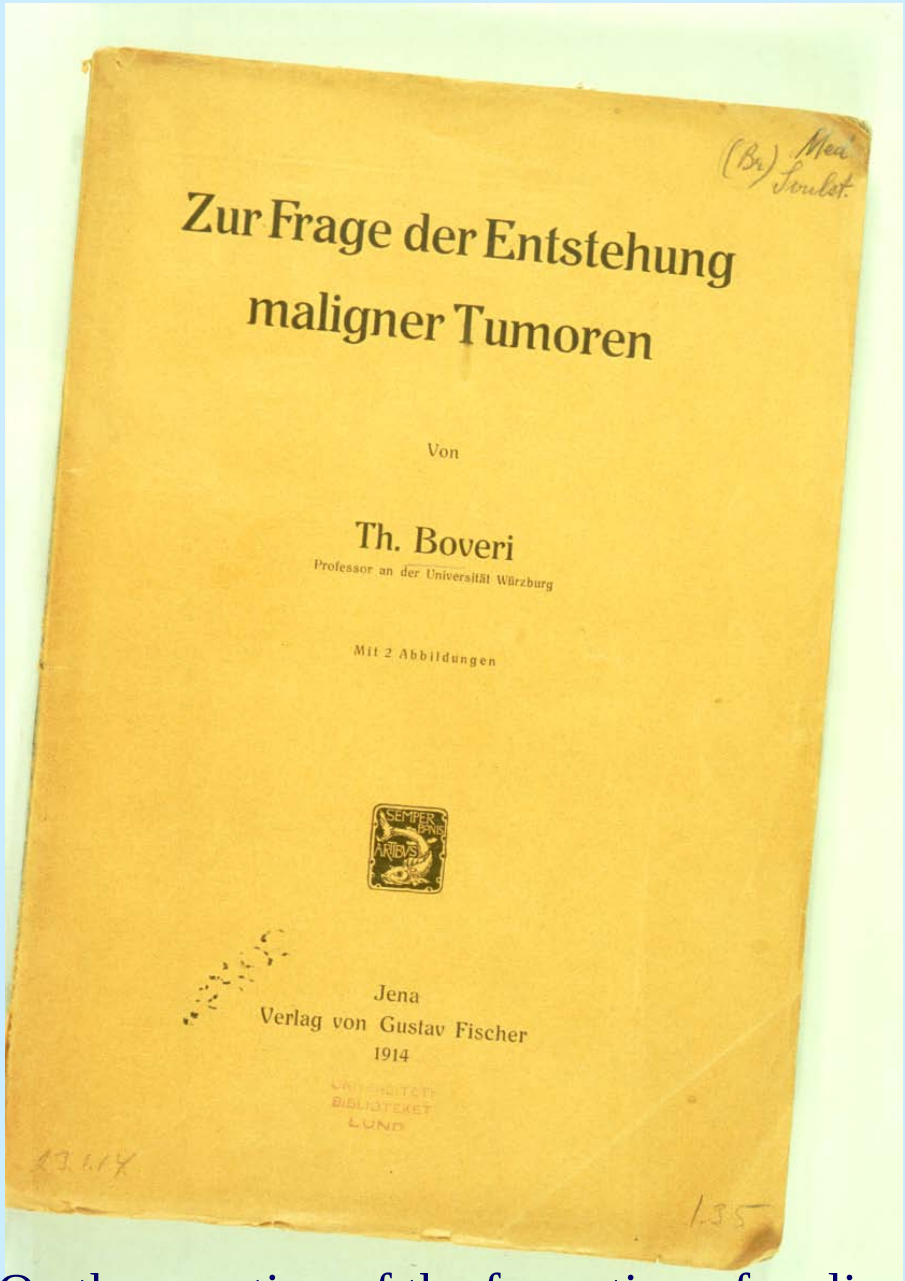
4 = 7-1-4/2

References

1. Bergeron A., *A very elementary presentation of the Hannenhalli-Pevzner theory*. Discrete Applied Mathematics, vol. 146, 134-145, 2001.
2. Marília D. V. Braga. *Exploring the solution space of sorting by reversals when analyzing genome rearrangements*. PhD thesis, University of Claude Bernard, 2009.
3. Guillaume Fertin, Anthony Labarre, Irena Rusu, Eric Tannier, Stephan Vialette. *Combinatorics of Genome Rearrangements*. The MIT Press,Cambridge, England, 2009.
4. ➔ Yancopoulos S., Attie O., Friedberg R. Efficient sorting of genomic permutations by translocation, inversion and block exchange. Bioinformatics 21, 3340 - 3346 2005.
5. ➔➔ Anne Bergeron, Julia Mixtacki, Jens Stoye. A unifying view of Genome Rearrangements. WABI 2006, LNBI 4175, 163-173, 2006.



Rearrangements in cancer



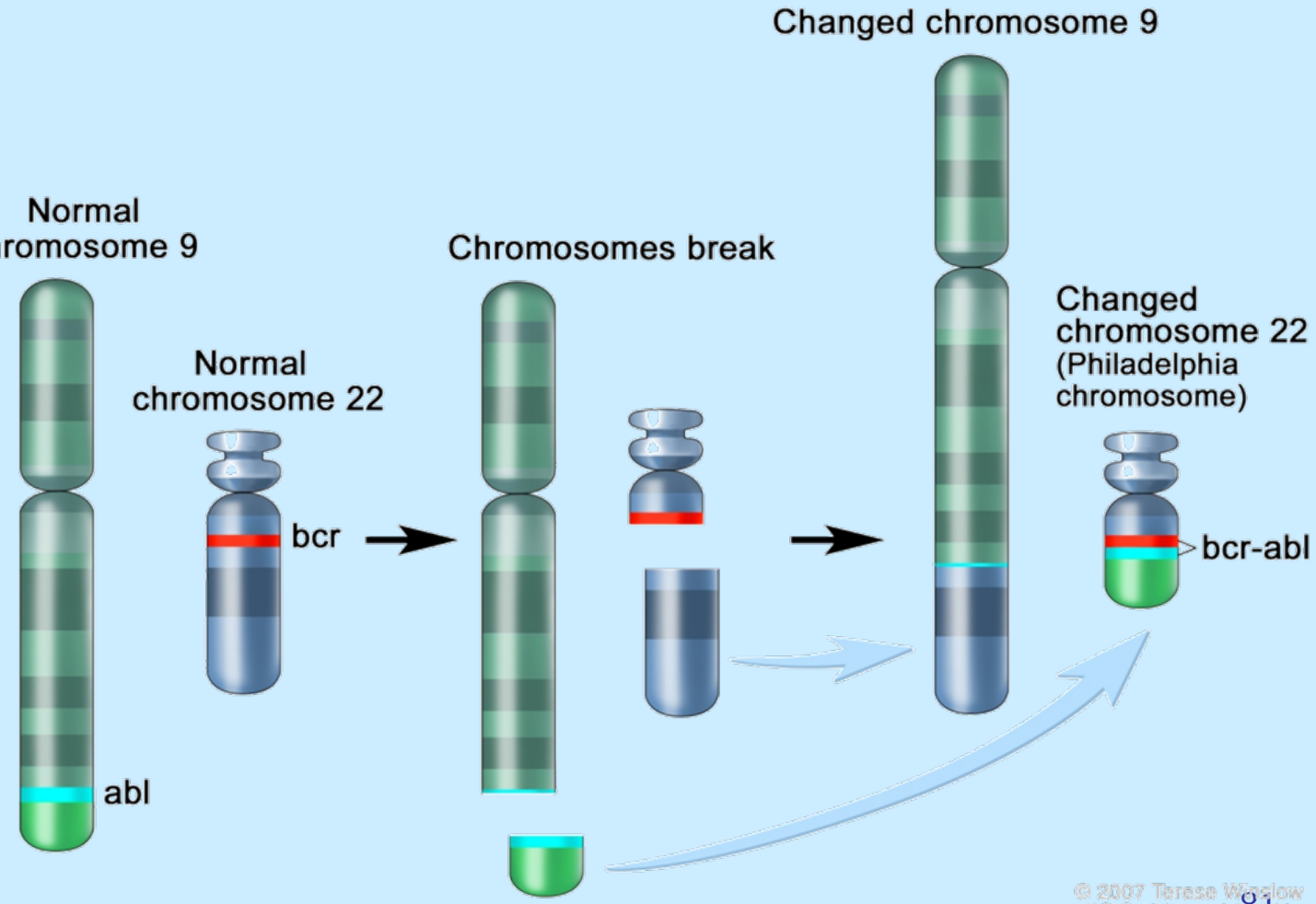
Theodor Boveri

(On the question of the formation of malignant tumors)

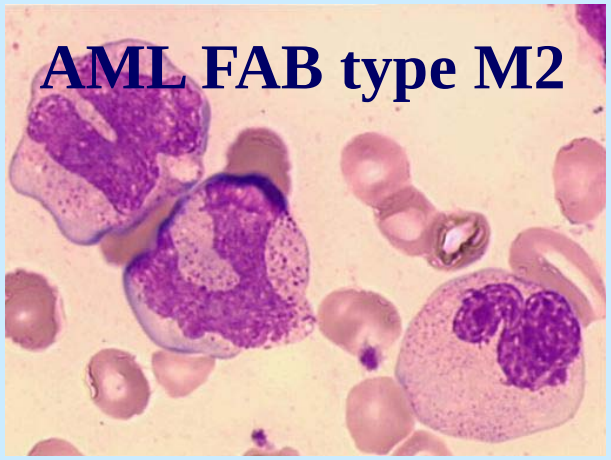
The "Philadelphia Chromosome"

Peter C. Nowell

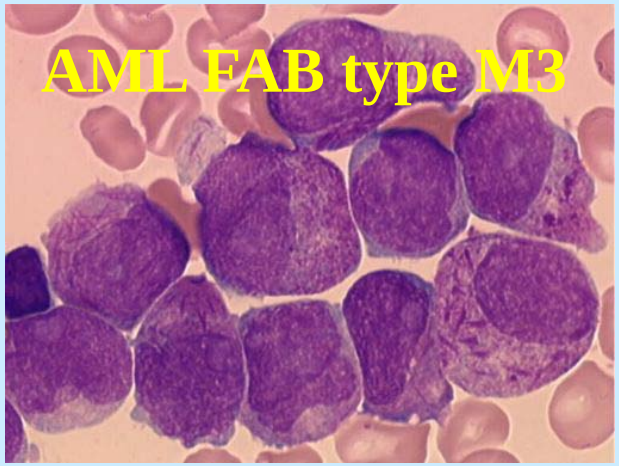
Janet D. Rowley



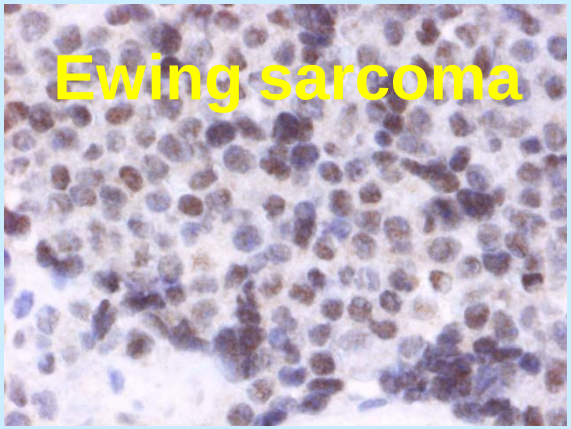
Chromosome Aberrations Typify Cancer Subtypes



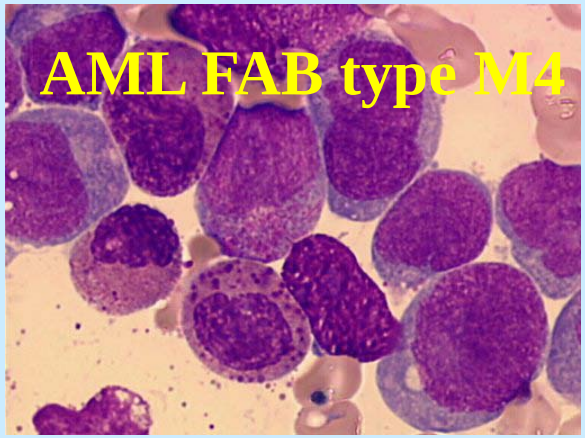
t(8;21)(q22;q22)



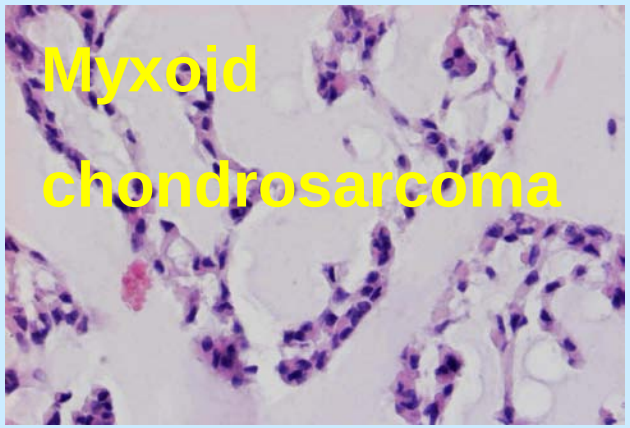
t(15;17)(q22;q21)



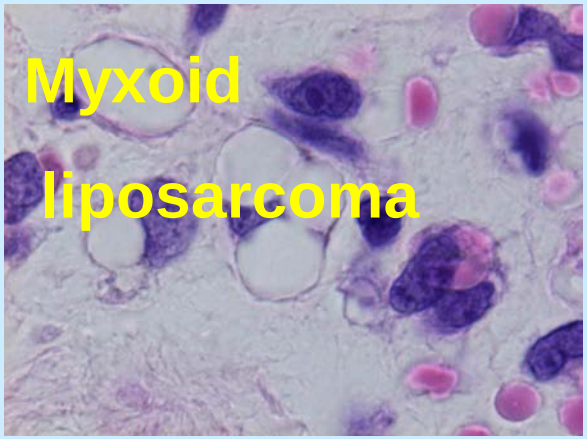
t(11;22)(q24;q12)



inv(16)(p13q22)



t(9;22)(q31;q12)

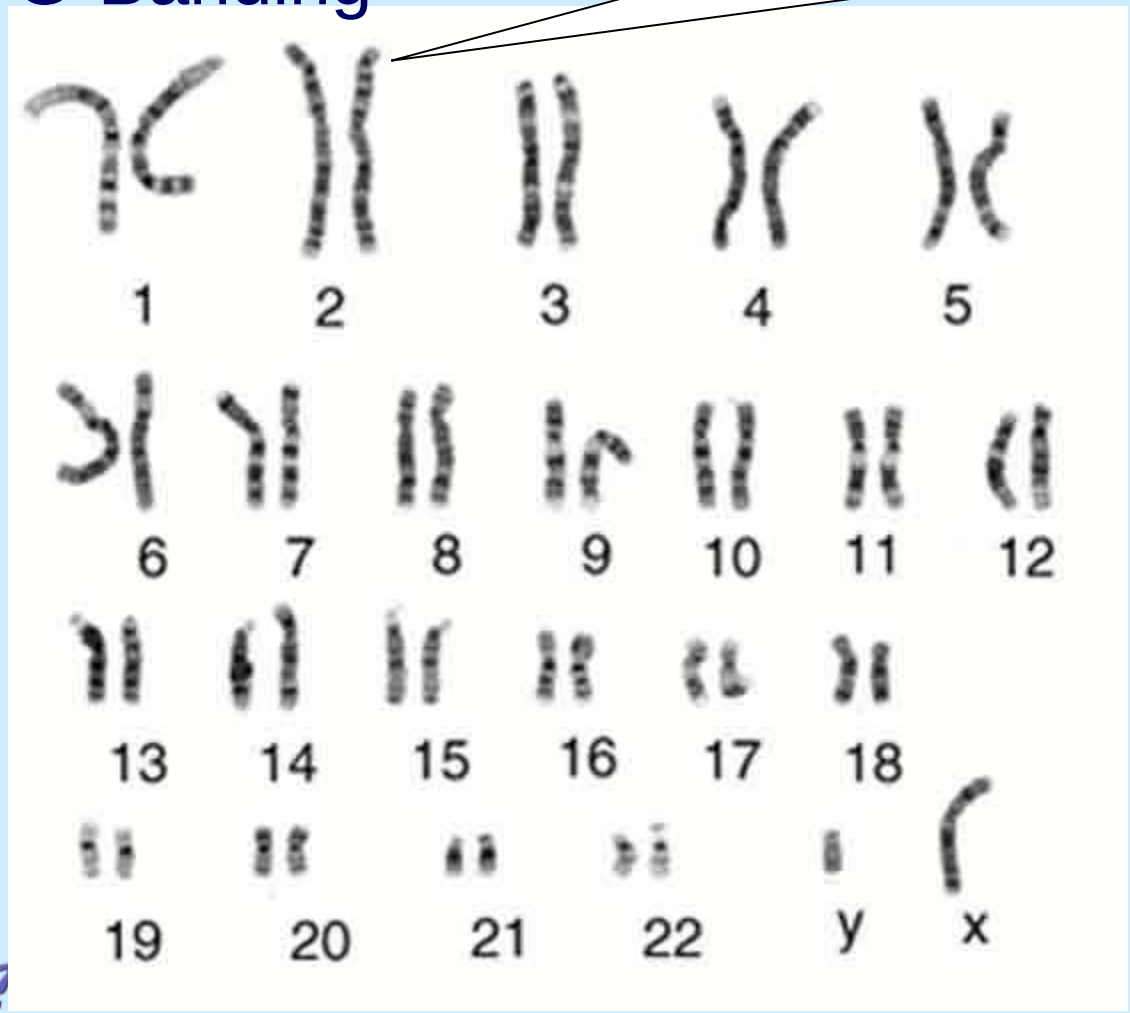


t(12;16)(q13;p11)

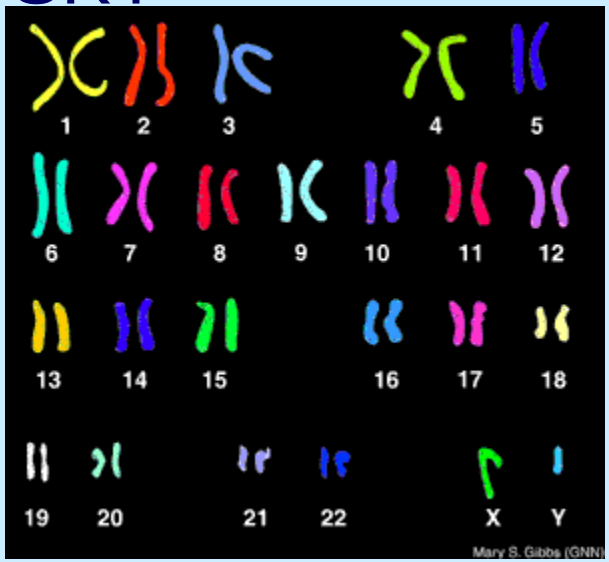
Karyotypes

G-Banding

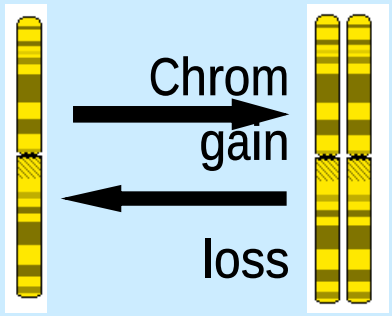
Bands resolution: 1 band ~ 5-10Mbp

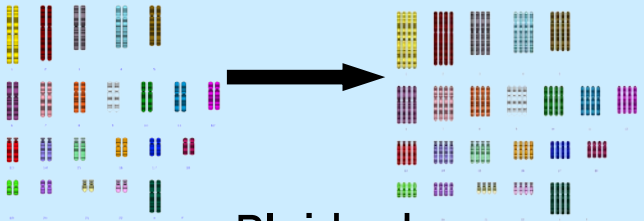


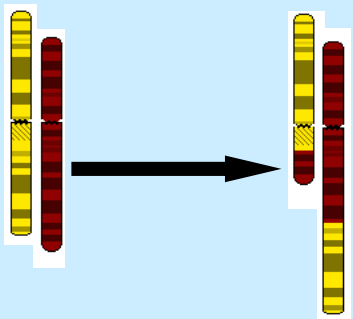
SKY

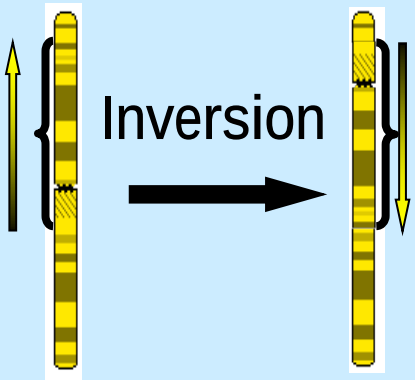


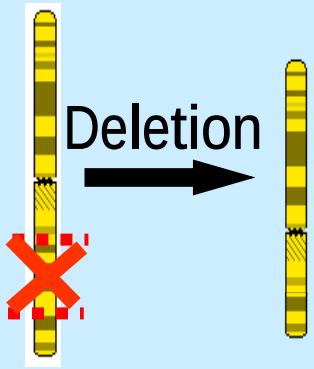
Events

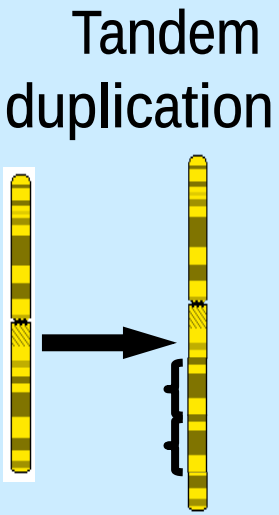

Chrom
gain
loss

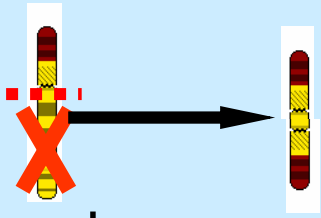

Ploidy change

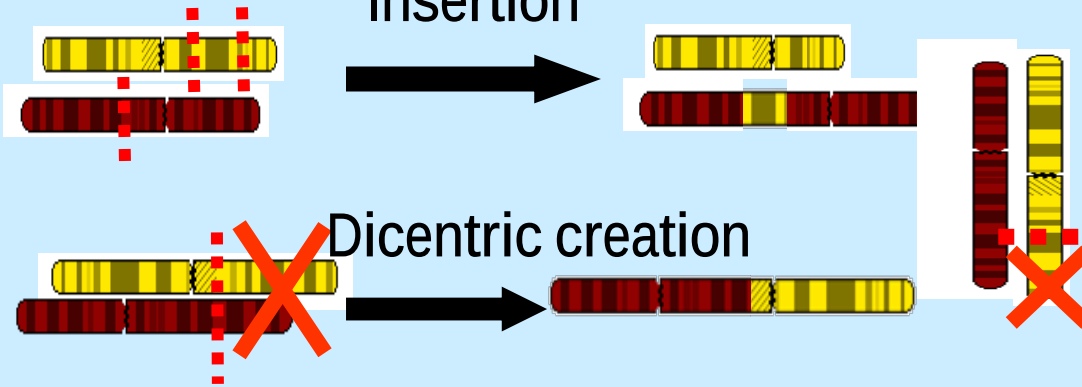

Translocation


Inversion

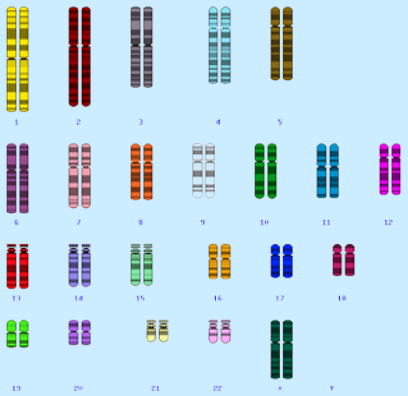

Deletion


Tandem
duplication

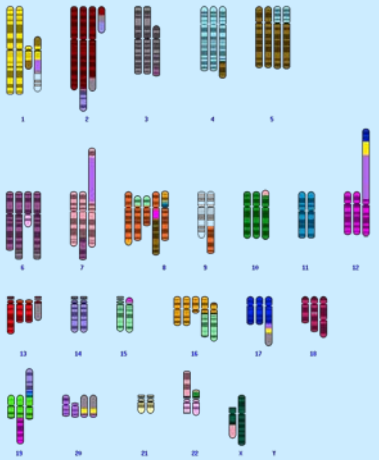

Iso-chromosome
creation


Insertion
Dicentric creation
Tai
duplicatio

The Karyotype Sorting Problem



Shortest sequece of events leading to the karyotype?



- Model with all operations seems intractable
- W developed a conservative heuristic
- Sorts uniquely 98% of >60K karyotypes in the Mitelman DB

FIN