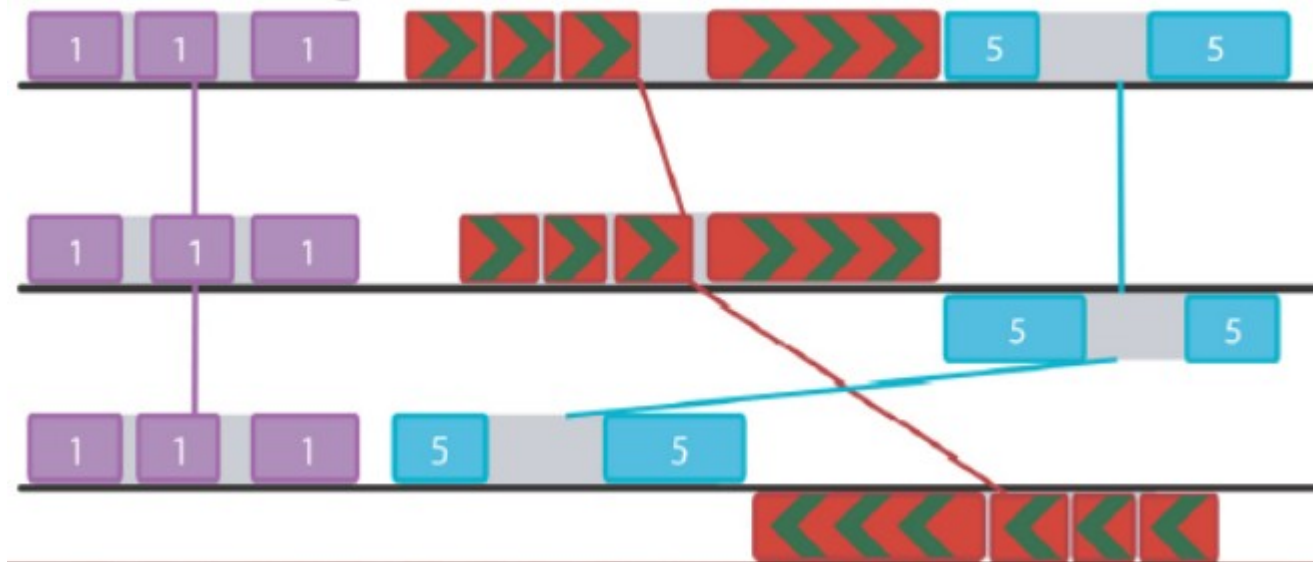


Aligning Genomes



Genome Analysis, 12 Nov 2007

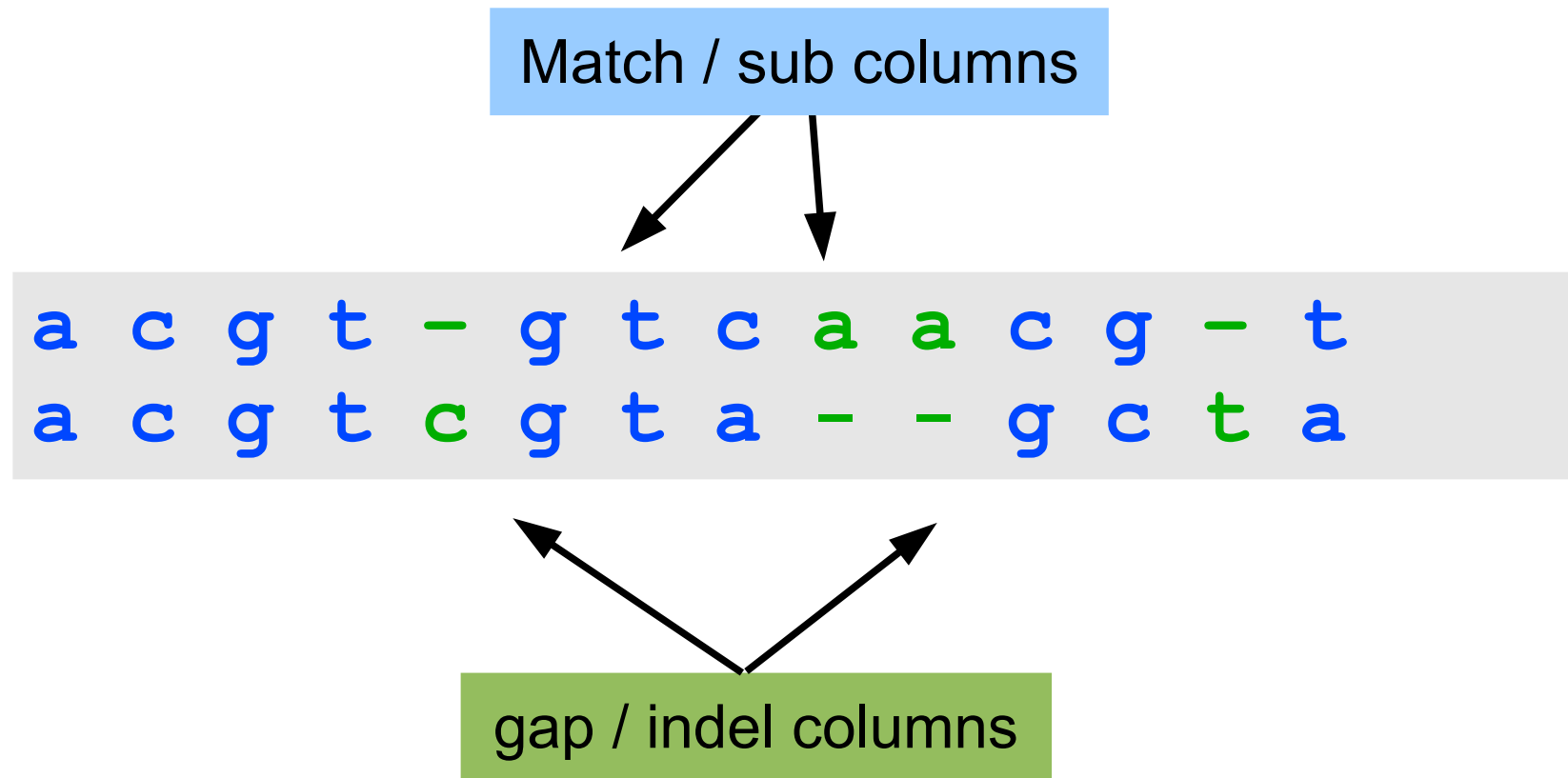
Several slides shamelessly stolen from Chr. Storm

Motivation

To compare sequences we typically first need to identify homologous segments.

This essentially means constructing an alignment of the sequences.

“Classical” local and global alignment



Pairwise global alignment

	a	t	c	t	c	c	a	c	c	
a	0	1	2	3	4	5	6	7	8	9
t	1	0	1	2	3	4	5	6	7	8
a	2	1	0	1	2	3	4	5	6	7
c	3	2	1	1	2	3	4	4	5	6
a	4	3	2	1	2	3	4	4	5	6
a	5	4	3	2	2	3	4	4	5	6
c	6	5	4	3	3	3	4	4	5	6
g	7	6	5	4	4	3	3	4	4	5
c	8	7	6	5	5	4	4	4	4	5
c	9	8	7	6	6	5	4	5	4	4

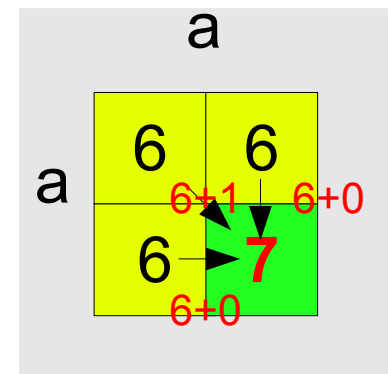
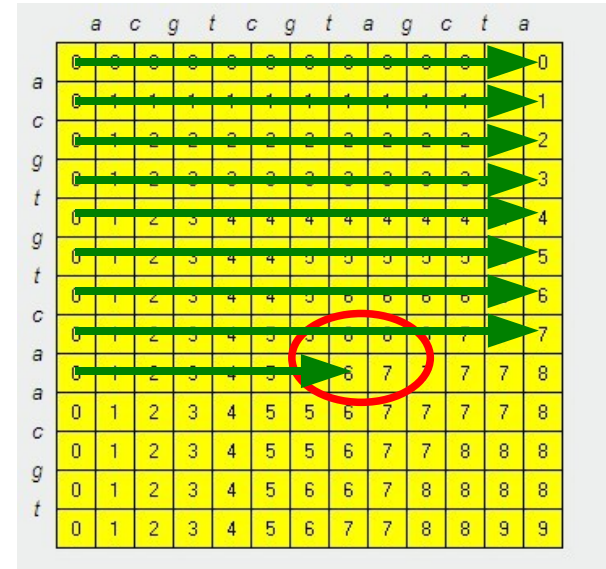
A global pairwise alignment is a path in the dynamic programming table.

Essentially similar for local alignments.

Heuristics needed for multiple alignments.

Pairwise global alignment

$$\text{Cost}(i, j) = \max \begin{cases} \text{Cost}(i-1, j-1) + \text{subcost}(A[i], B[j]) \\ \text{Cost}(i-1, j) + \text{gapcost} \\ \text{Cost}(i, j-1) + \text{gapcost} \\ 0 \text{ if } i=0 \text{ and } j=0 \end{cases}$$



Pairwise global alignment

Backtracking to obtain alignment from DP table.

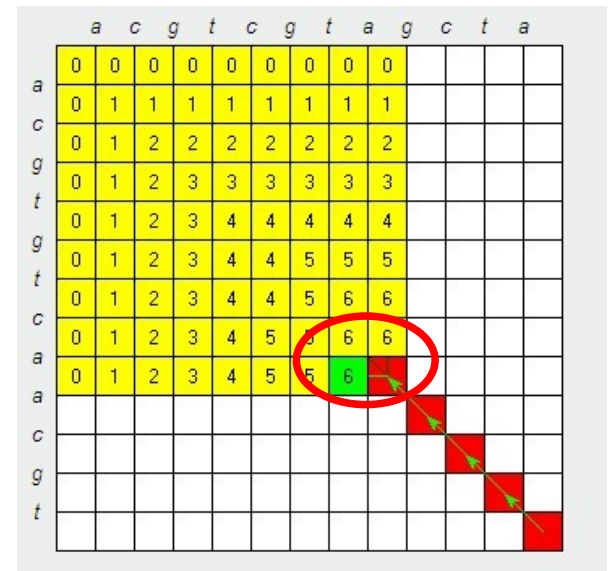
Space and runtime complexity $O(nm)$ with n length of first sequence and m length of second sequence.

	a	c	g	t	c	g	t	a	g	c	t	a
a	0	0	0	0	0	0	0	0	0			
c	0	1	1	1	1	1	1	1	1			
g	0	1	2	2	2	2	2	2	2			
t	0	1	2	3	3	3	3	3	3			
c	0	1	2	3	4	4	4	4	4			
g	0	1	2	3	4	4	5	5	5			
t	0	1	2	3	4	4	5	6	6			
a	0	1	2	3	4	5	5	6	6			
a	0	1	2	3	4	5	5	6	6			
c												
g												
t												

Pairwise global alignment

Backtracking to obtain alignment from DP table.

Space and runtime complexity $O(nm)$ with n length of first sequence and m length of second sequence.



For the biologists: $O(nm)$ means “order of” or proportional to n times m

Pairwise global alignment

Backtracking to obtain alignment from DP table.

	a	c	g	t	c	g	t	a	g	c	t	a
a	0	0	0	0	0	0	0	0	0			
c	0	1	1	1	1	1	1	1	1			
c	0	1	2	2	2	2	2	2	2			

Say we want to align:

Human Chr 22 ($\approx 50,000,000$ bp) vs. Mouse Chr 19 ($\approx 60,000,000$ bp)

Running time:

$50,000,000 \times 60,000,000 \text{ op} / 100,000,000 \text{ op/sec} = 347 \text{ days}$

For the biologists: $O(nm)$ means “order of” or proportional to n times m

Heuristics

... the art of the possible ...

Try to detect and report as many biological reasonable similarities within reasonable time (and space) ...

Banding

... the art of the possible ...

Try to detect and report as many biological reasonable similarities within reasonable time (and space) ...

Observation:

Realistic alignments do not stray much from the diagonal of the DP table.

Banding

	a	c	g	t	c	g	t	a	g	c	t	a
a	0	0	0	0	0	0	0	0	0	0	0	0
c	0	1	1	1	1	1	1	1	1	1	1	1
g	0	1	2	2	2	2	2	2	2	2	2	2
t	0	1	2	3	3	3	3	3	3	3	3	3
a	0	1	2	3	4	4	4	4	4	4	4	4
c	0	1	2	3	4	5	5	5	5	5	5	5
g	0	1	2	3	4	5	6	6	6	6	6	6
t	0	1	2	3	4	5	6	7	7	7	7	7
a	0	1	2	3	4	5	6	7	8	8	8	8
c	0	1	2	3	4	5	6	7	8	8	8	8
g	0	1	2	3	4	5	6	7	8	8	8	8
t	0	1	2	3	4	5	6	7	8	8	9	9

Banding

Time and space is now $O(b \cdot \max\{n, m\})$ where b is the band width.

	g	t	a	g	c	t	a
g	0	0	0	0	0	0	0
	1	1	1	1	1	1	1
	2	2	2	2	2	2	2
	3	3	3	3	3	3	3

Consider again

Human Chr 22 ($\approx 50,000,000$ bp) vs. Mouse Chr 19 ($\approx 60,000,000$ bp)
with, say, band width 5000

Running time:

$5000 \times 60,000,000 \text{ op} / 100,000,000 \text{ op/sec} = 3000 \text{ sec} = 50 \text{ min}$

Banding

Time and space is now $O(b \cdot \max\{n, m\})$ where b is the band width.

	g	t	a	g	c	t	a
g	0	0	0	0	0	0	0
	1	1	1	1	1	1	1
	2	2	2	2	2	2	2
	3	3	3	3	3	3	3

Consider again

Human Chr 22 ($\approx 50,000,000$ bp) vs. Mouse Chr 19 ($\approx 60,000,000$ bp)
with, say, band width 500

Running time:

$5000 \times 60,000,000 \text{ op} / 100,000,000 \text{ op/sec} = 3000 \text{ sec} = 50 \text{ min}$

But beware: Banding is *very* sensitive to large (or a high number of) indels.

Anchors

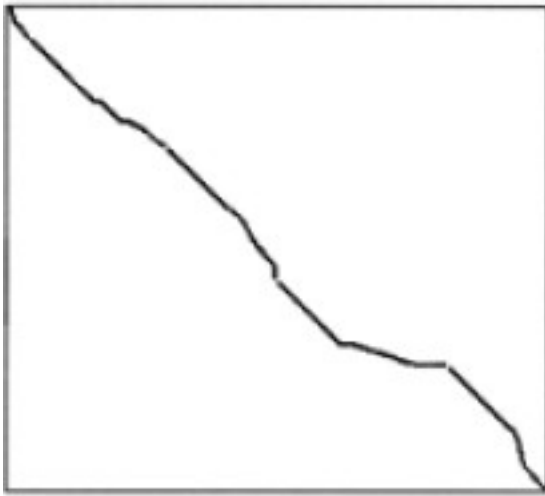
... the art of the possible ...

Try to detect and report as many biological reasonable similarities within reasonable time (and space) ...

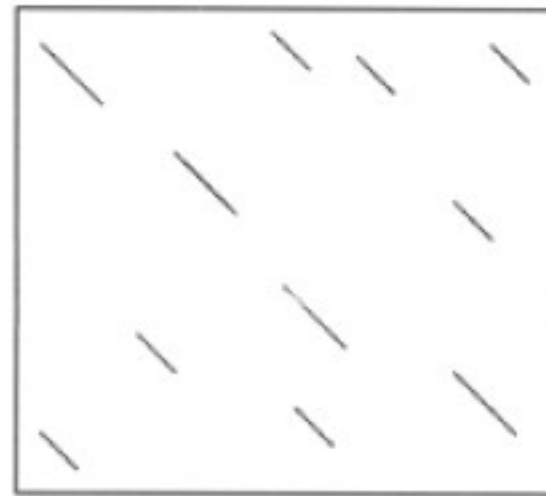
Observation:

Highly similar sub-sequences will be part of a realistic alignment

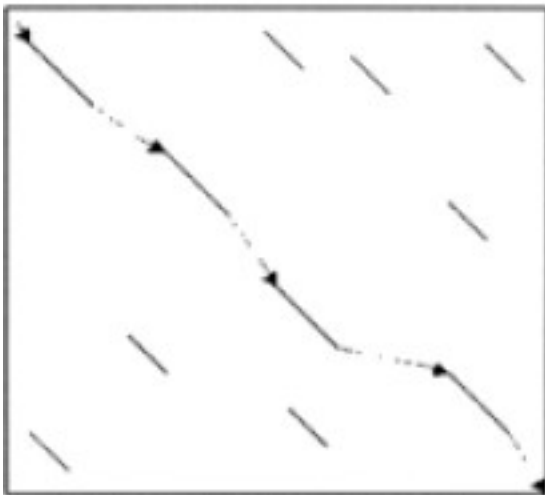
Anchors



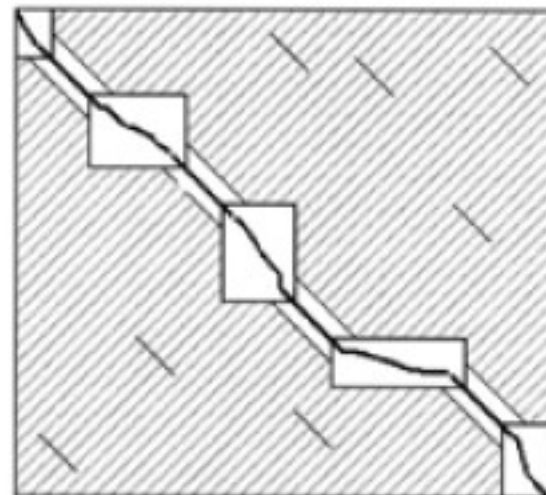
A



B

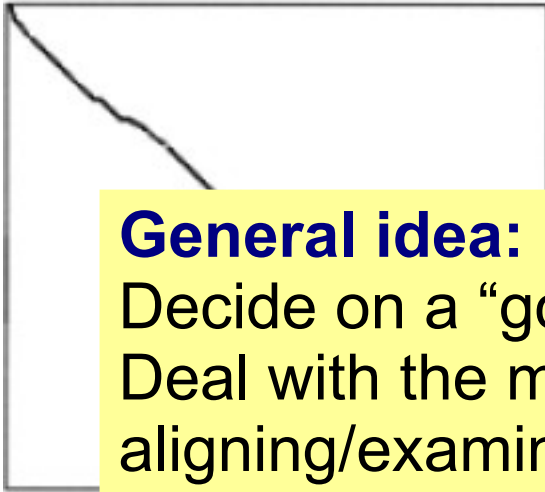


C

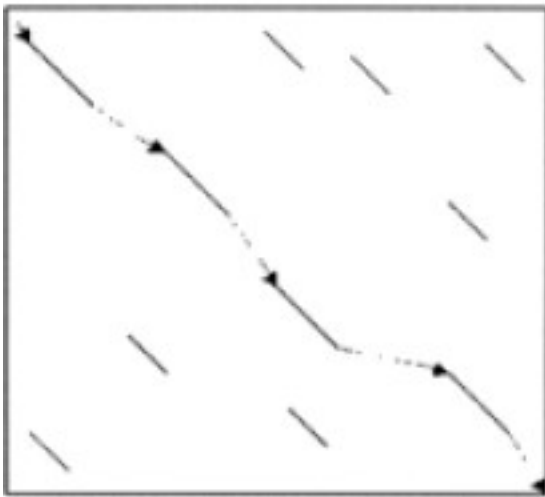


D

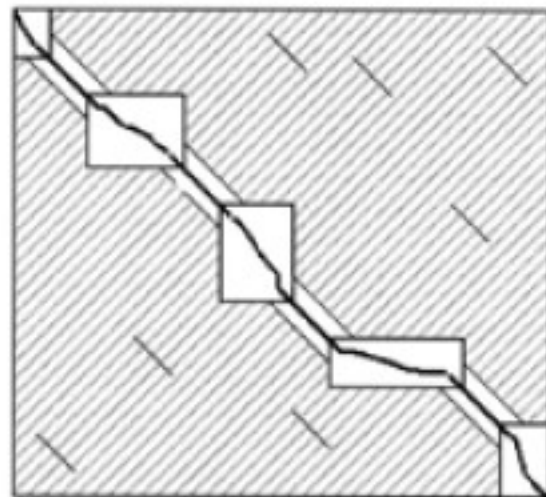
Anchors



General idea: Find “large scale” similarities. Decide on a “good” selection of these to align. Deal with the much smaller subproblems of aligning/examining “the rest” independently ...



C



D

Finding anchors

Finding the anchors is a local alignment problem – but the Smith-Waterman algorithm will of course not work.

Several approaches to solving this problem – with either exact or approximate anchor matching.

Next few slides show one of these approaches; one that finds exact matching anchors.

Maximal Unique Match

MUMs are sequences in genomes *A* and *B* that:

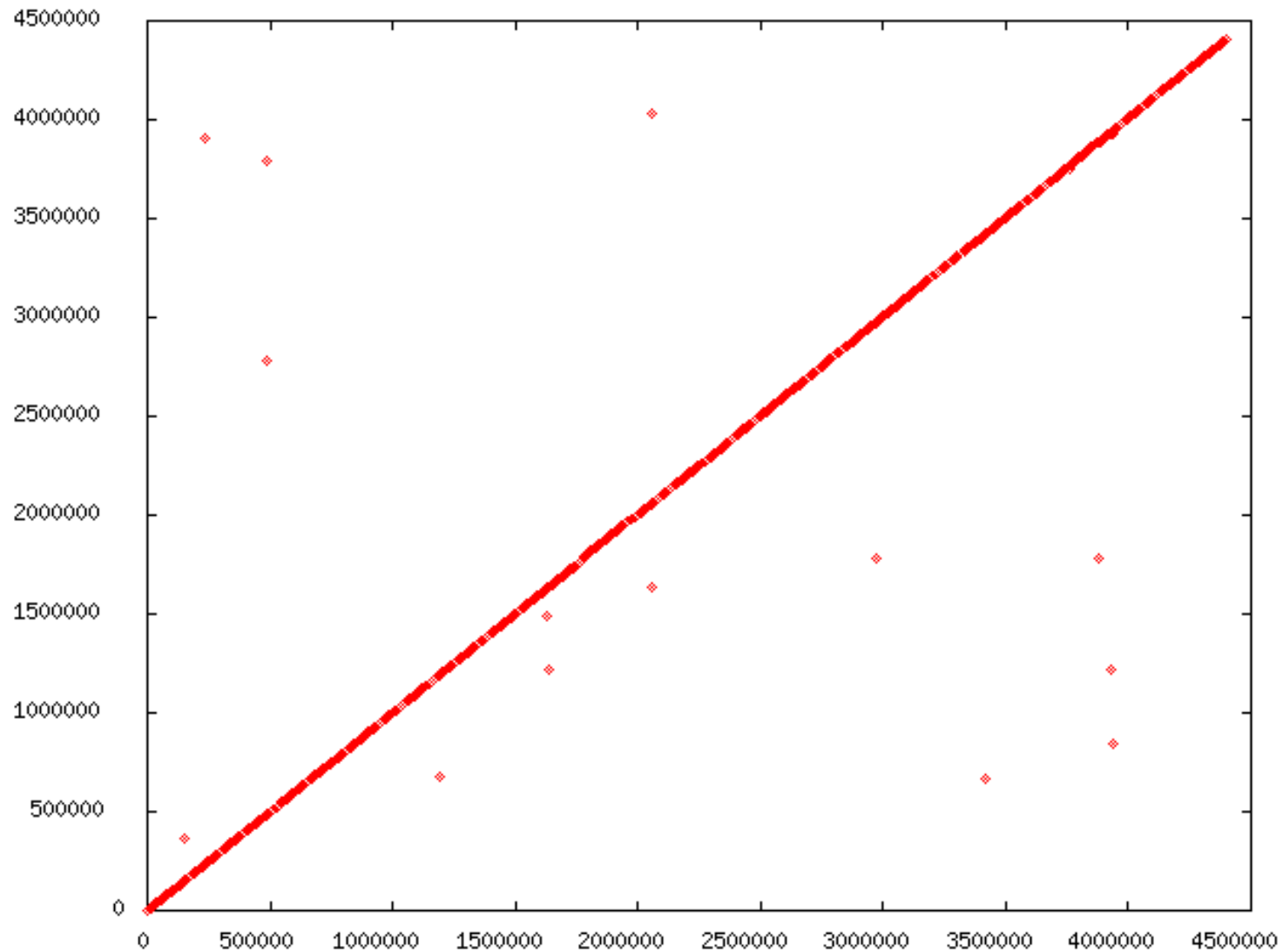
Occur *exactly once* in *A* and in *B*

Are not contained in any larger matching sequence



Genome *A*: tcgata**GACGATCGCGCCGTAGATCGAATAACGAGAGAGCATAA**cgactta
Genome *B*: gcatta**GACGATCGCGCCGTAGATCGAATAACGAGAGAGCATAA**tccagag

M. tuberculosis CDC1551 vs H37rV

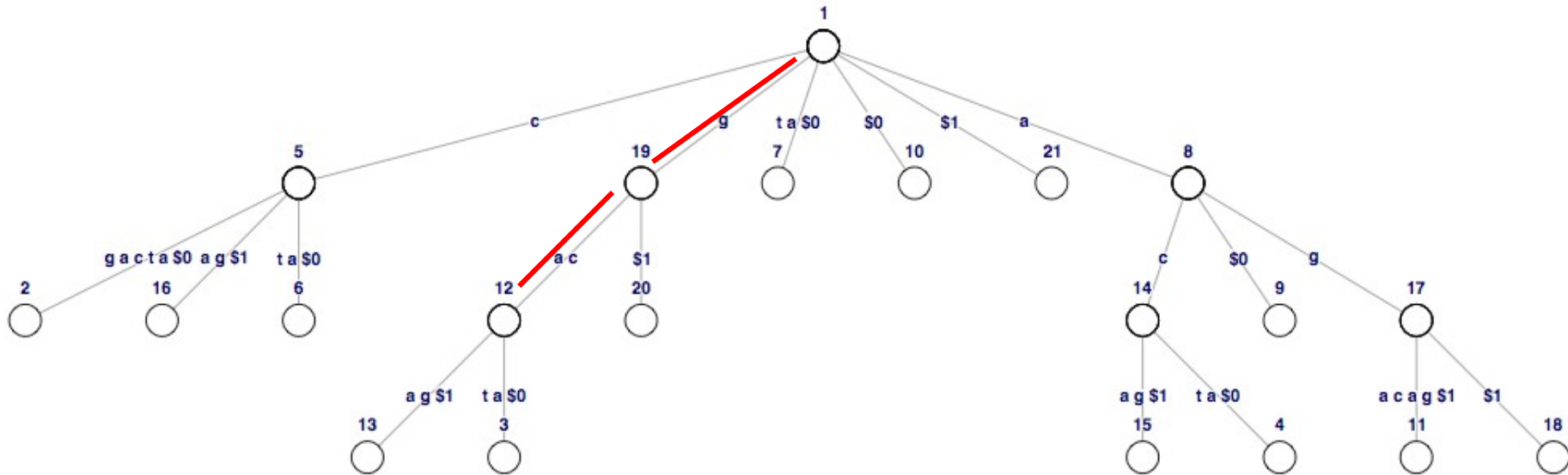


Finding MUMs

MUMs can efficiently be found using *suffix trees*...

MUMs in a (generalized) suffix tree

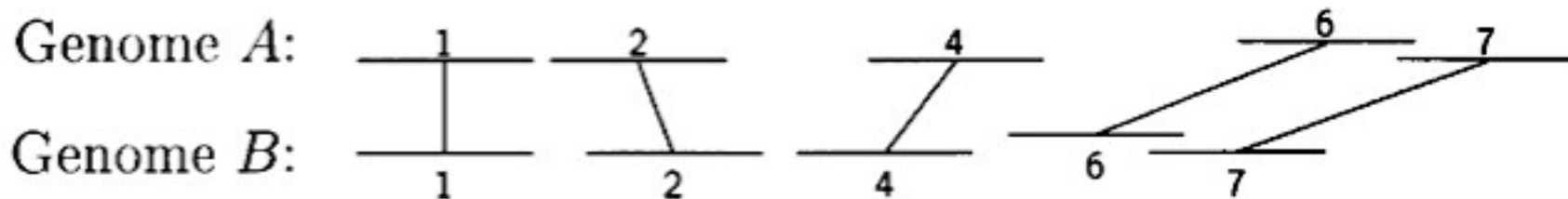
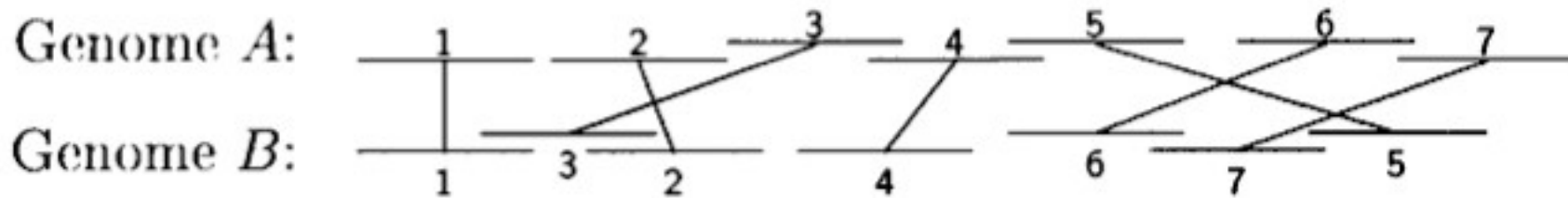
Consider *cgacta* and *agacag*, contains *gac* as a MUM



... a MUM is the path-label of a node with exactly two child nodes that are leaf nodes from each genome. This implies uniqueness and right-maximality. Left-maximality can be checked by lookup in the genomes. Total time is $O(n+m)$, the size of the suffix tree ...

Putting the MUMs together

Find the Longest Increasing Sequence

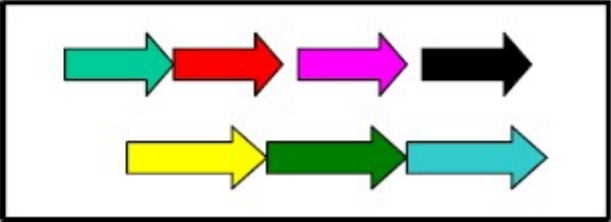


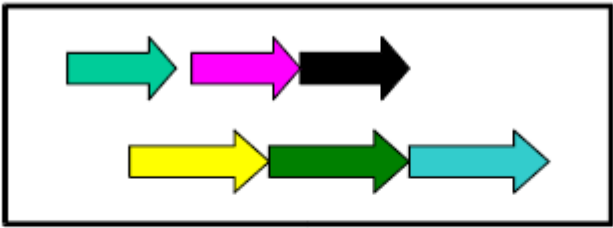
Easy to solve in time $O(k^2)$ using dynamic programming (can be done in time $O(k \log k)$) where k is the number of MUMs.

Co-linearity

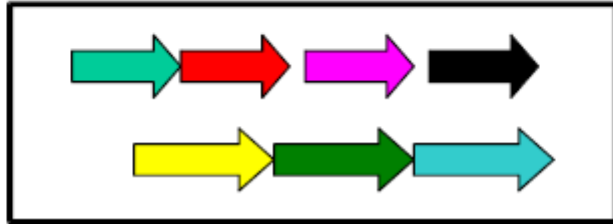
All these algorithms consider only *co-linear* sequences.

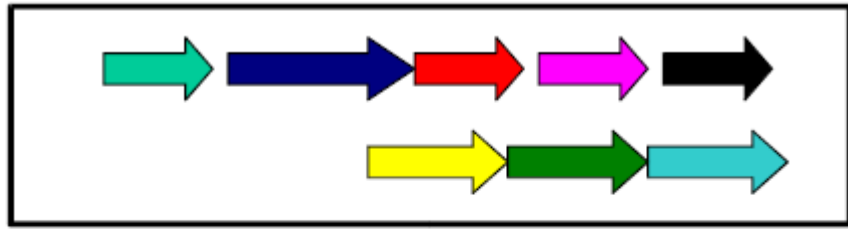
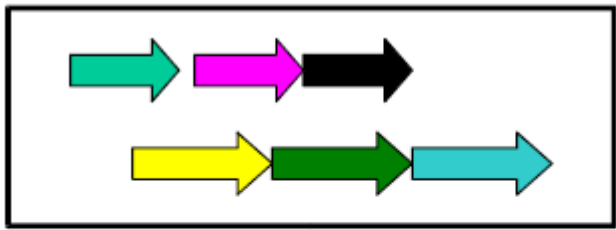
But only *very* closely related species have co-linear genomes!





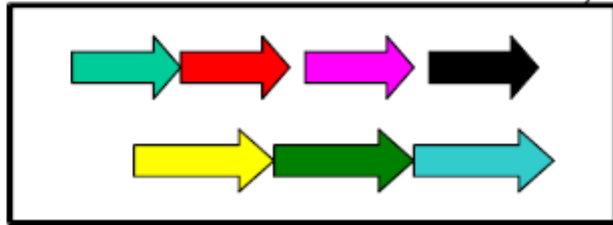
deletion

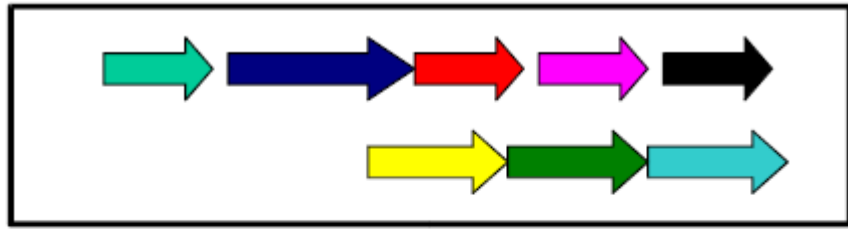
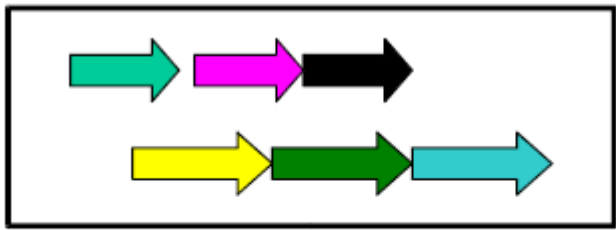




deletion

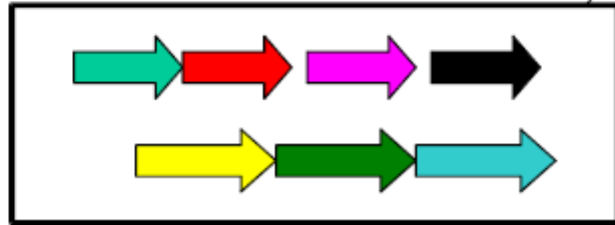
insertion



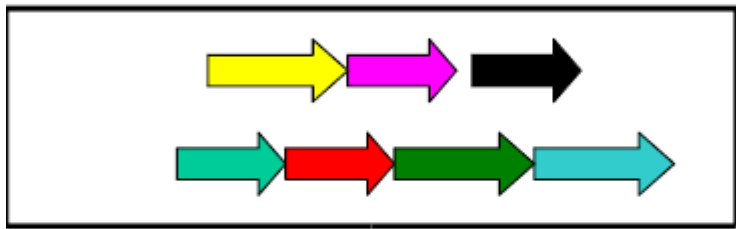


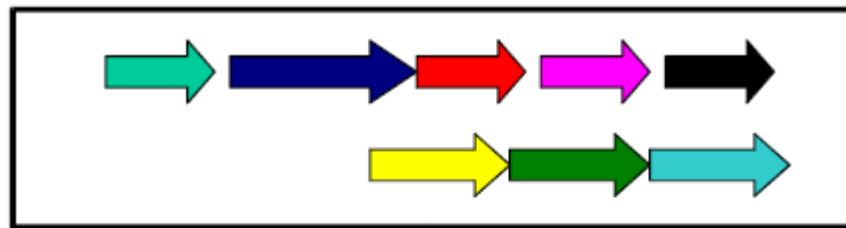
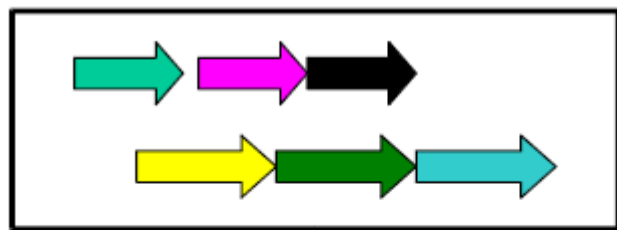
deletion

insertion



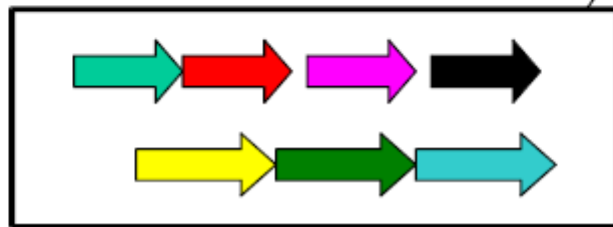
translocation





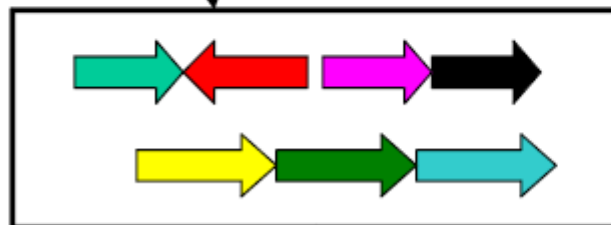
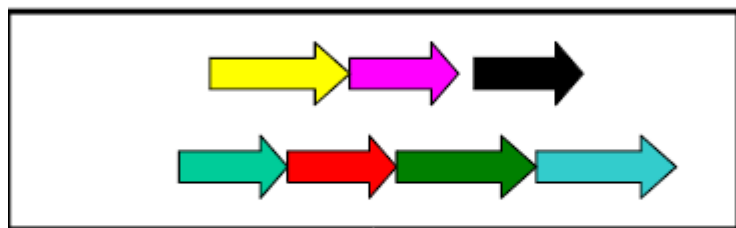
deletion

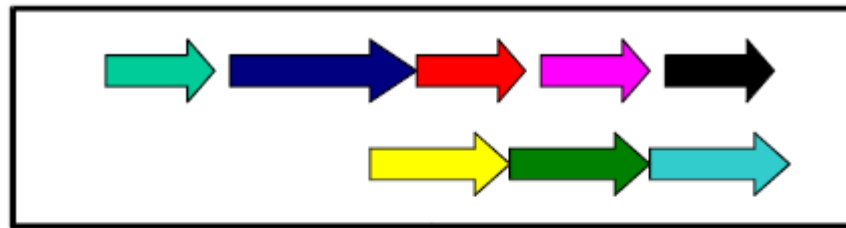
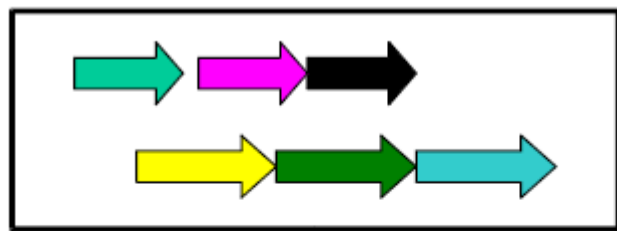
insertion



translocation

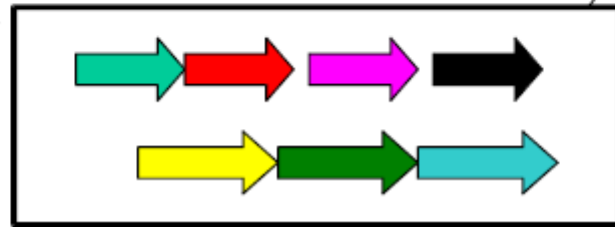
inversion





deletion

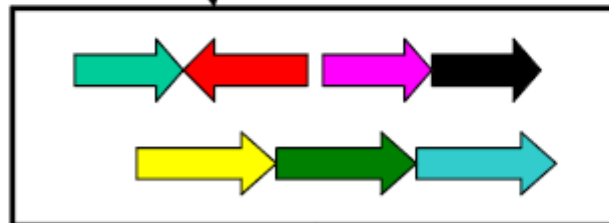
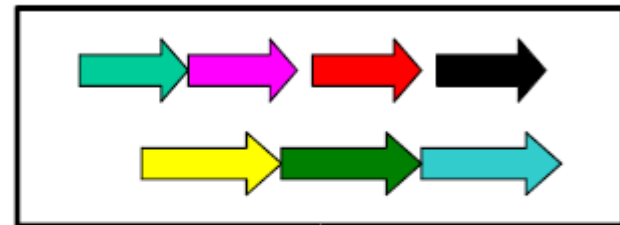
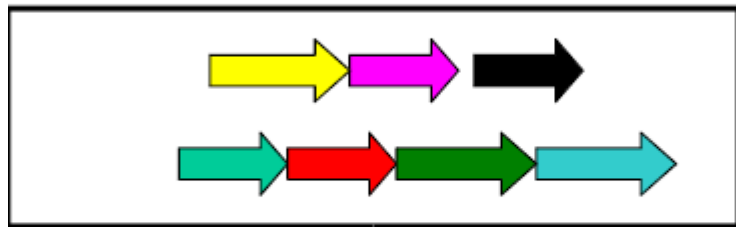
insertion



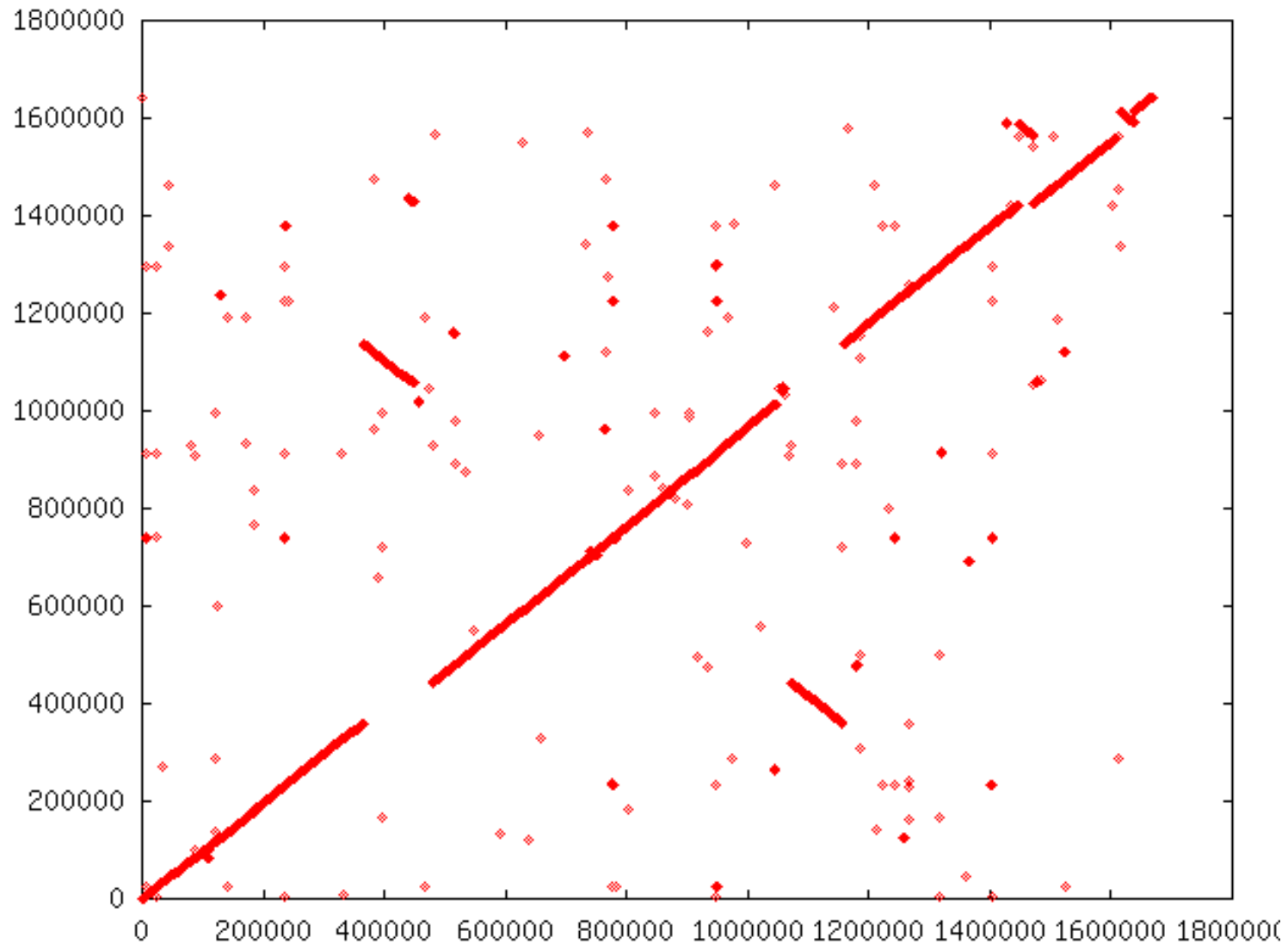
transposition

translocation

inversion

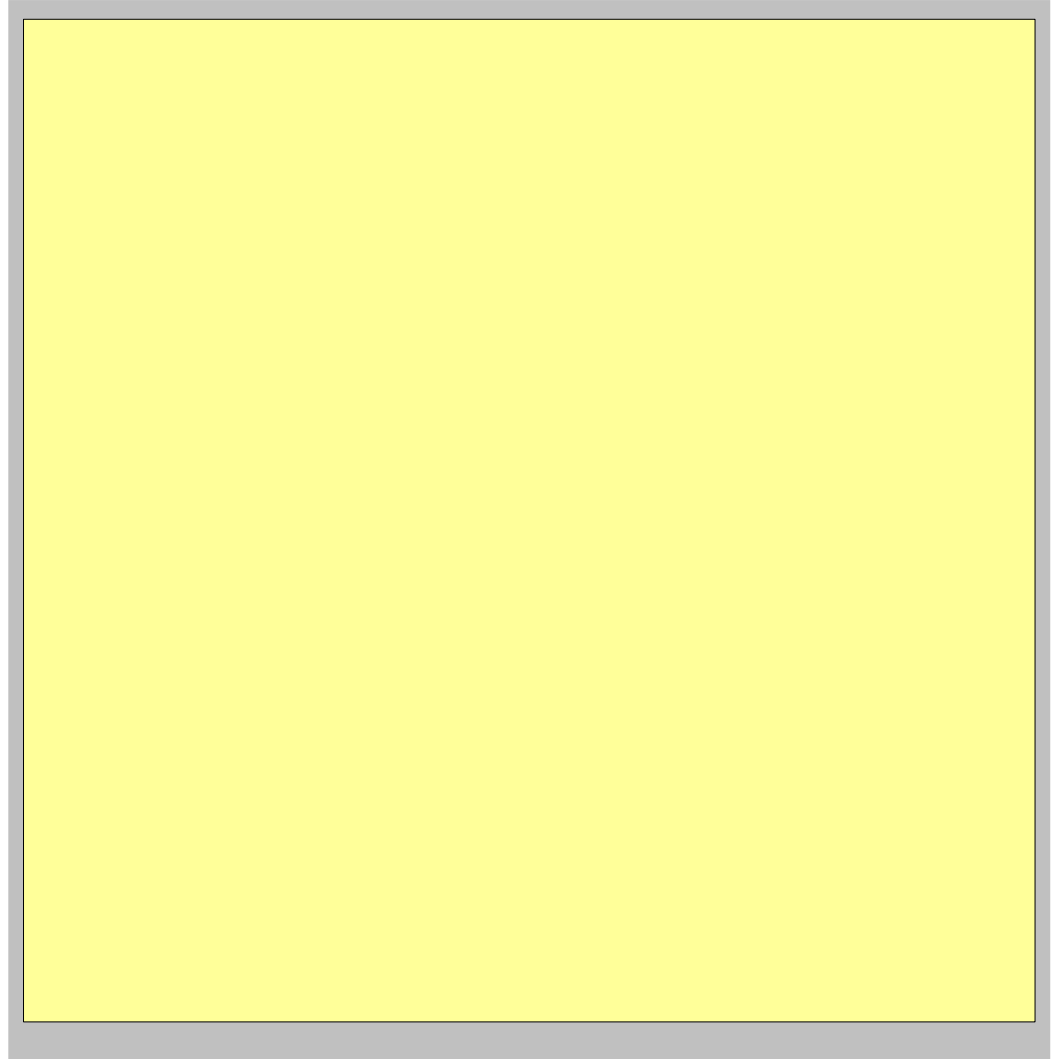


Helicobacter pylori strain 26695 vs J99



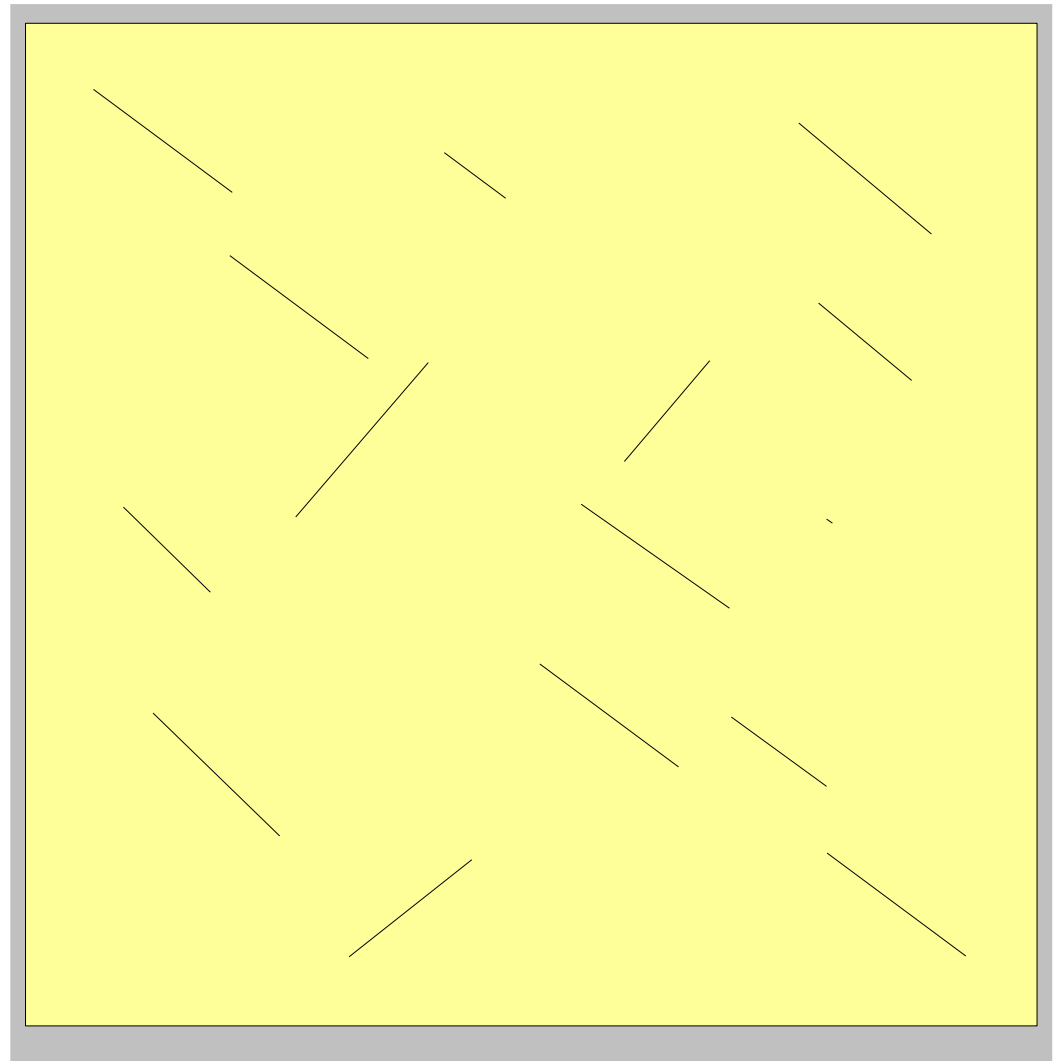
Dealing with rearrangements

Need an initial step for recognizing co-linear blocks that can then be aligned (heuristically or exact with the dynamic programming method).



Dealing with rearrangements

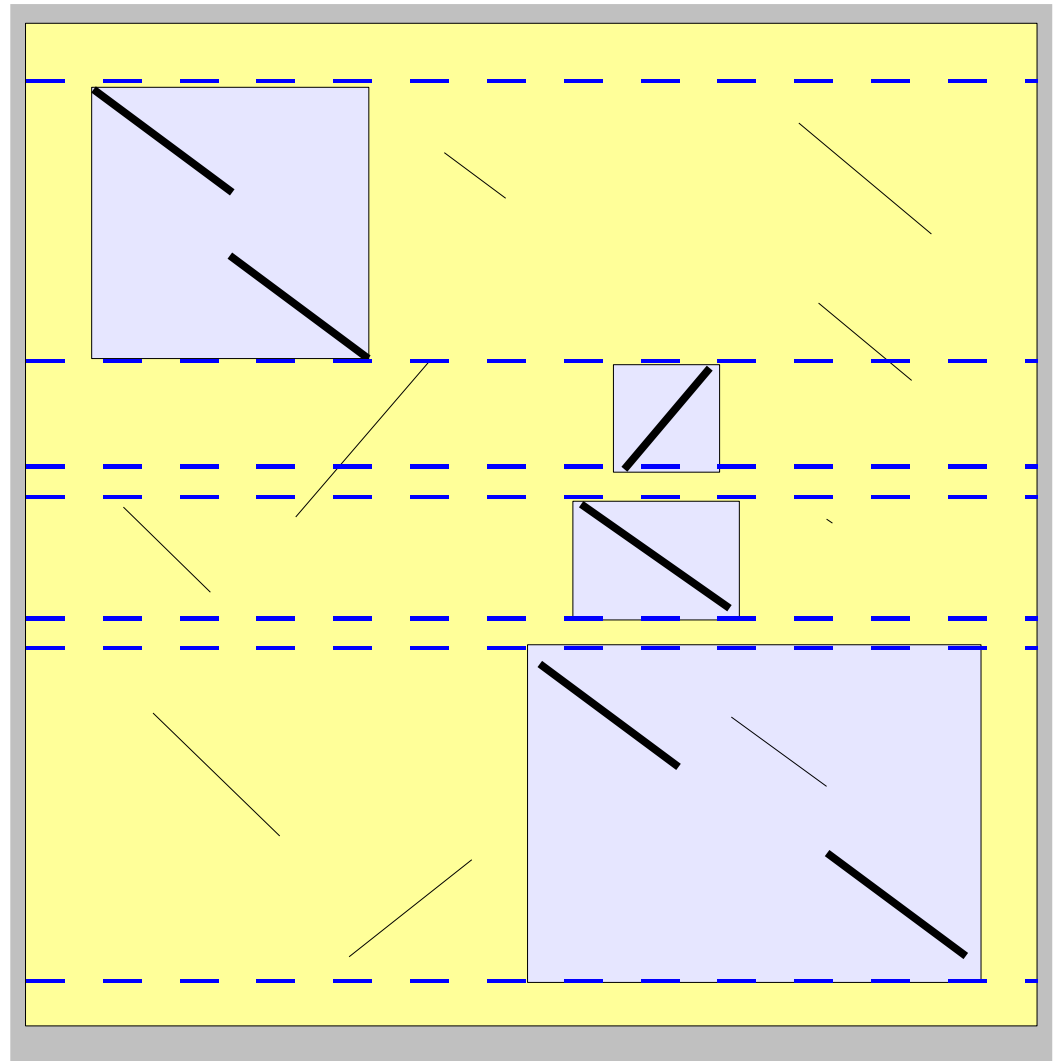
Locate “high scoring”
local alignments
(anchors).



Dealing with rearrangements

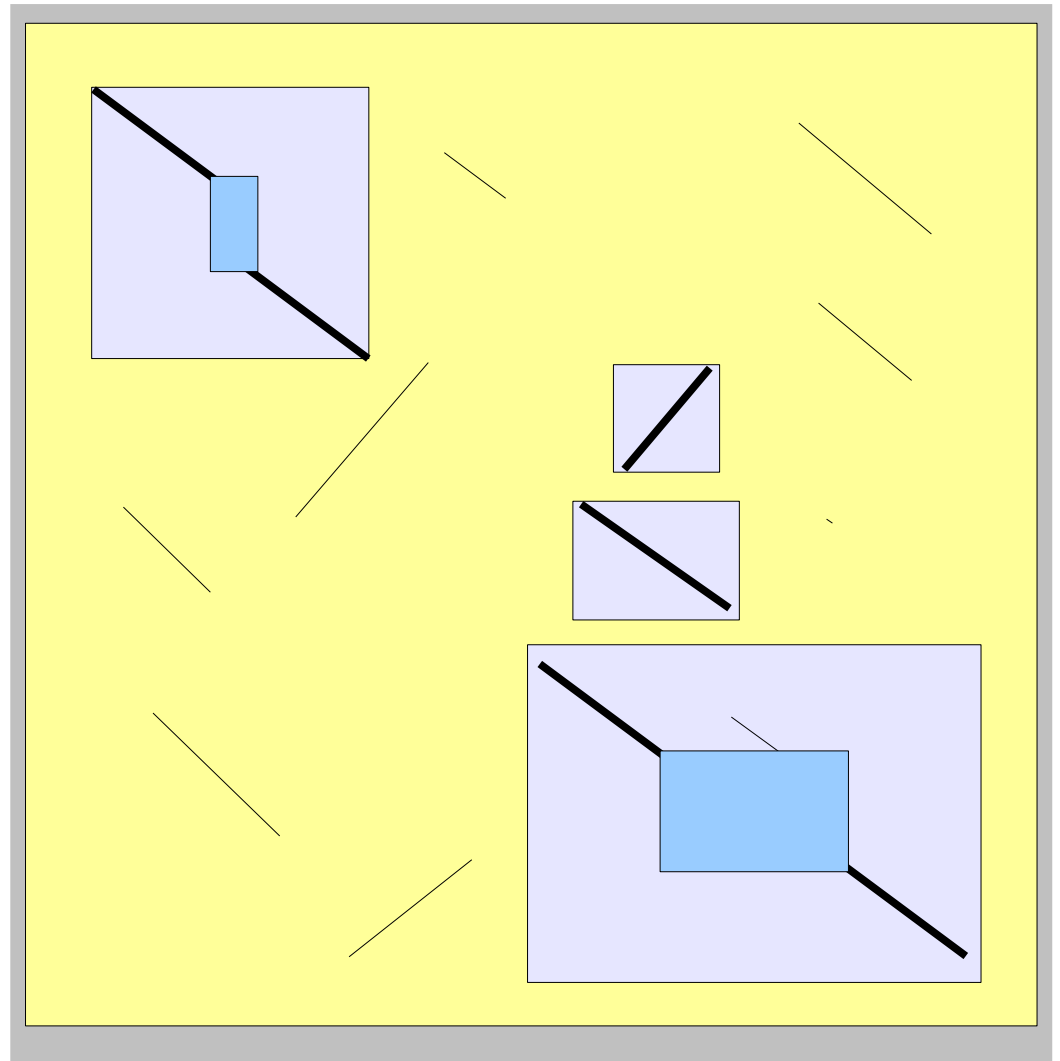
Identify larger likely co-linear blocks by combining local alignments

Possibly excluding aligning the same nucleotide more than once in either one or both sequence



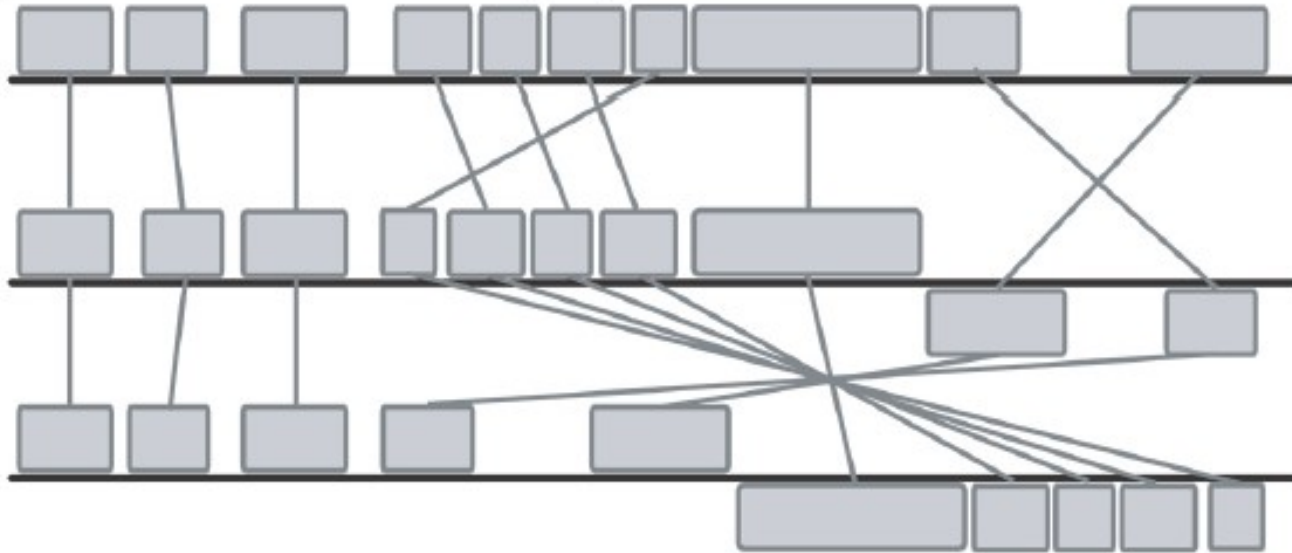
Dealing with rearrangements

Non-anchor parts of co-linear blocks can be handled recursively or through dynamic programming alignments.

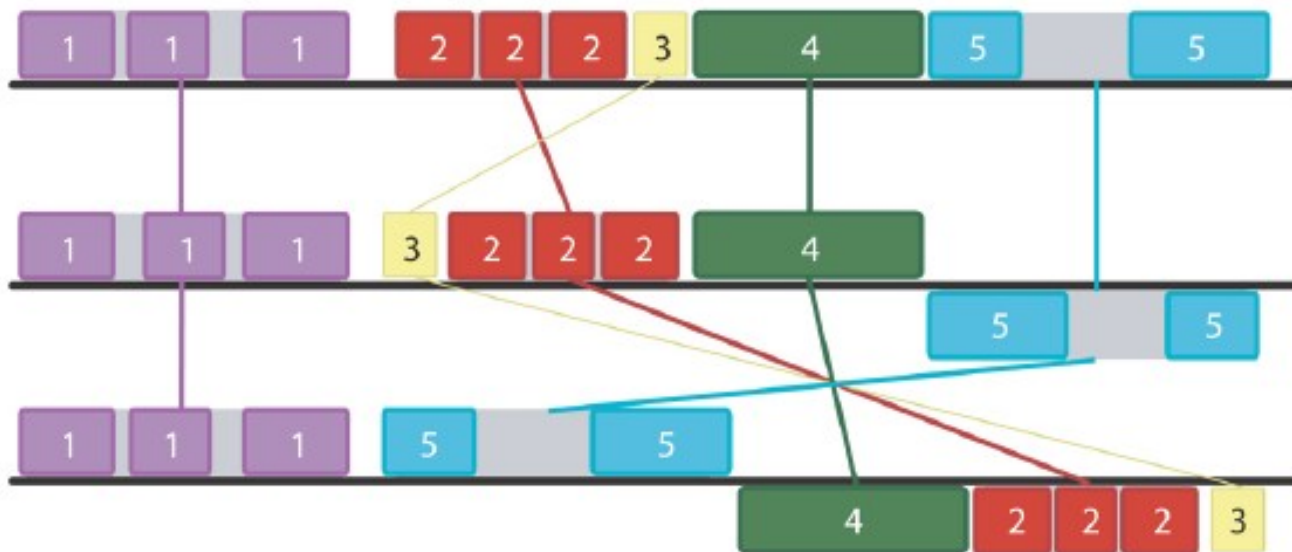


Example

A) The initial set of matching regions:

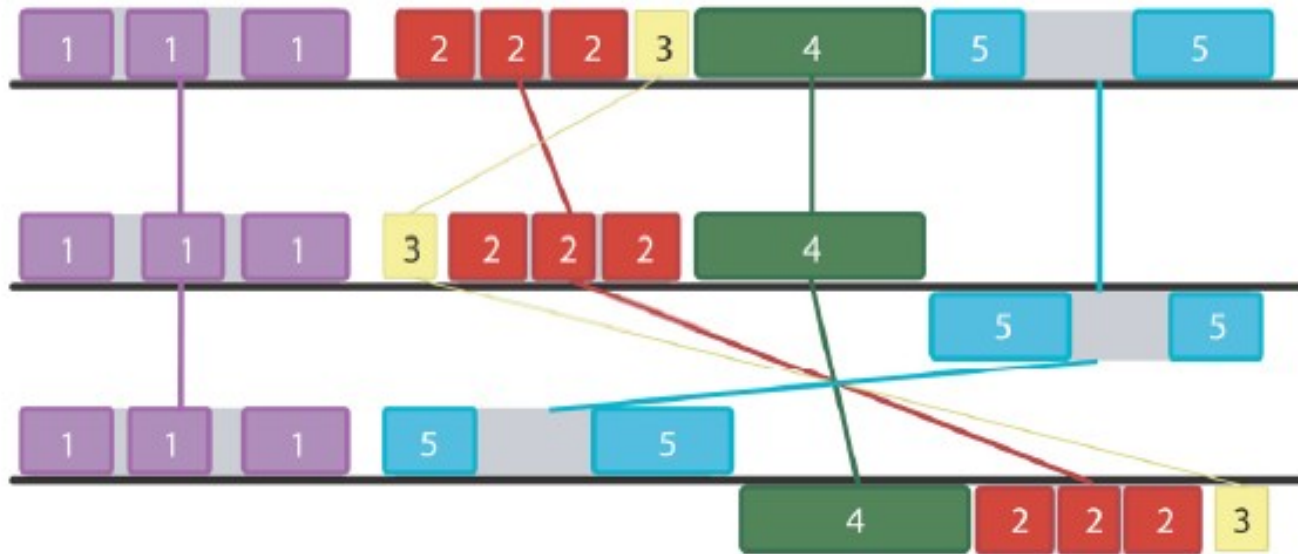


B) Minimum partitioning into collinear blocks:

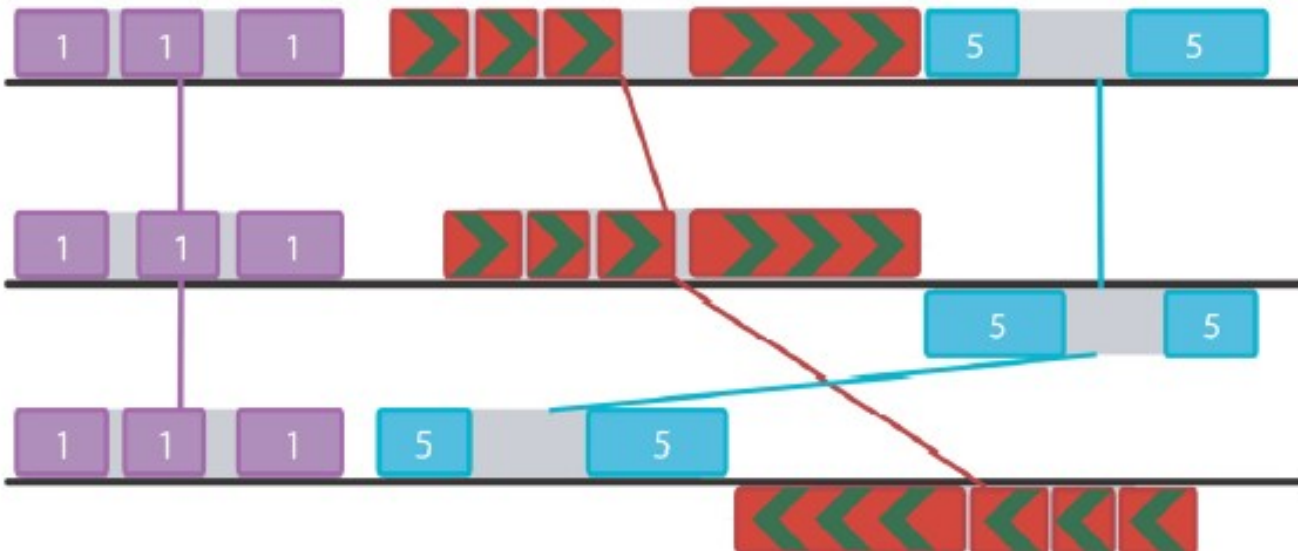


Example

B) Minimum partitioning into collinear blocks:



C) After removing block 3:



Multiple sequence alignments

The dynamical programming approach doesn't scale to multiple sequences (and several heuristics has the same problem).

Multiple sequence alignments

The dynamical programming approach doesn't scale to multiple sequences (and several heuristics has the same problem).

Yet again heuristics are needed!

Multiple sequence alignments

A common approach is *progressive alignment* where sequences are aligned pairwise along a “guide tree”

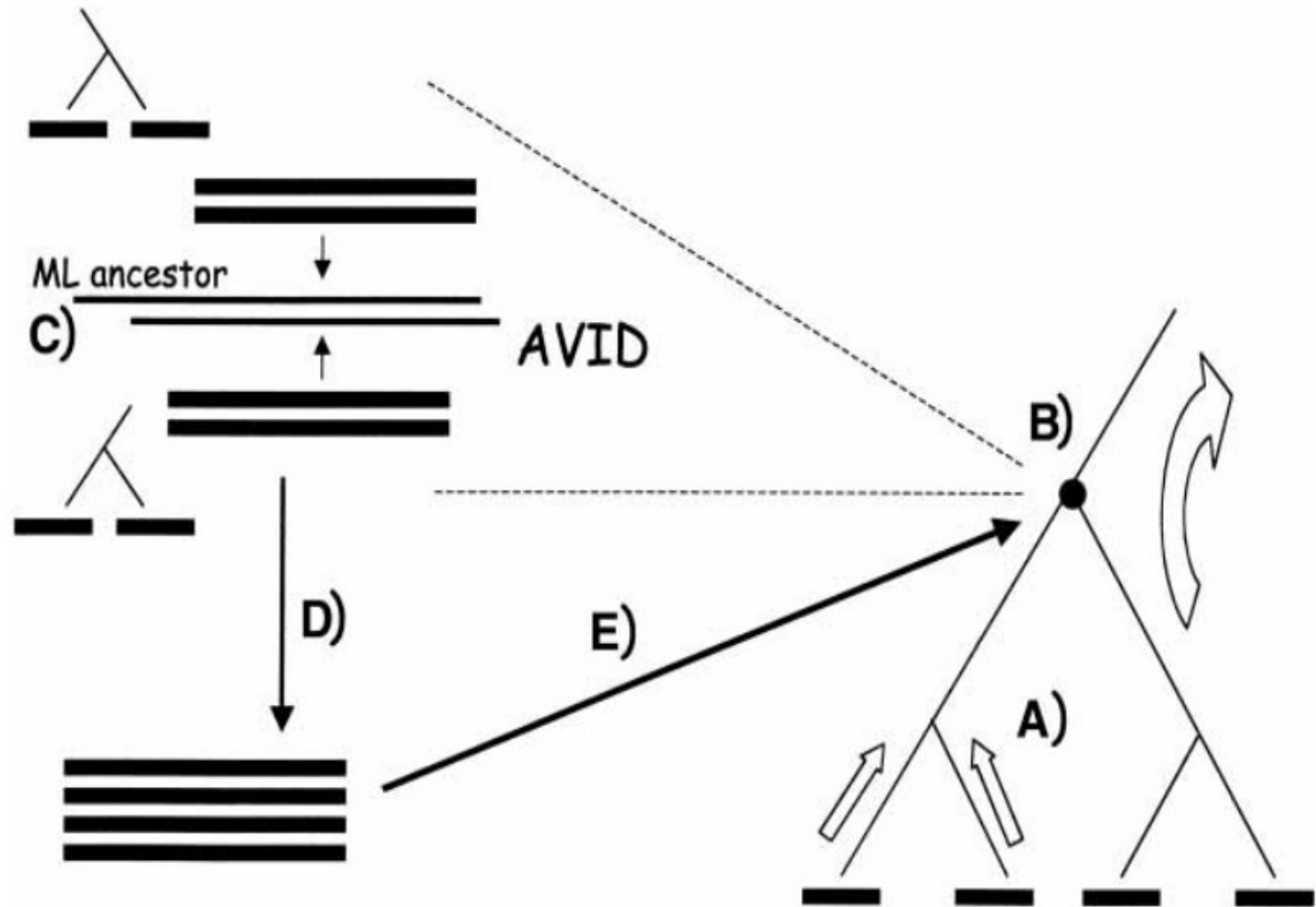


Figure 1 MAVID architecture overview. (A) Sequences are aligned upward along a guide tree and (B) alignments of alignments are performed at internal nodes. To align two alignments (C), maximum likelihood ancestor sequences are inferred from each of the separate alignments, and (D) the ancestor sequences are aligned with MAVID. The resulting multiple alignment (E) (corresponding to a subset of leaves of the tree) is then recorded at the internal node.

Summary

Structural rearrangements and the large size of genomes complicates aligning them.

Heuristics are needed to

- Identify co-linear segments
- Align large segments
- Handle multiple sequences