Sequence Alignment

Sequence alignment

- Introduction sequence alignment
- Calculation of an alignment
  - exercises
- Different types of the algorithm
  - exercises
- Wrap-up

Sequence alignment

- **Introduction sequence alignment**
- Calculation of an alignment
  - exercises
- Different types of the algorithm
  - exercises
- Wrap-up

Sequence comparison

Pairwise sequence alignment

Multiple sequence alignment

Database search

What is a pairwise alignment?

Two sequences are placed opposite each other to show similarity between the two

Insertions, deletions and mutations are introduced in both sequences to achieve this

```
CTCCTGAGGCAAATCTGTCAGTCCATCCTGGCTGAGTCCTCGCAGTCCCCGGCAGATCTTGAAGAAAAGA
||||||||| ||| ||| |||||  |  ||||| |  |   ||||  | | ||   ||| || ||| |||
CTCCTGAGGAAAACCTGCCAGTCAGTTGTGGCTAATCCTCAGCAGCACTCAGC---TCTGGAGGAAGAGA
```

Biological relevance

Insights into gene and protein function

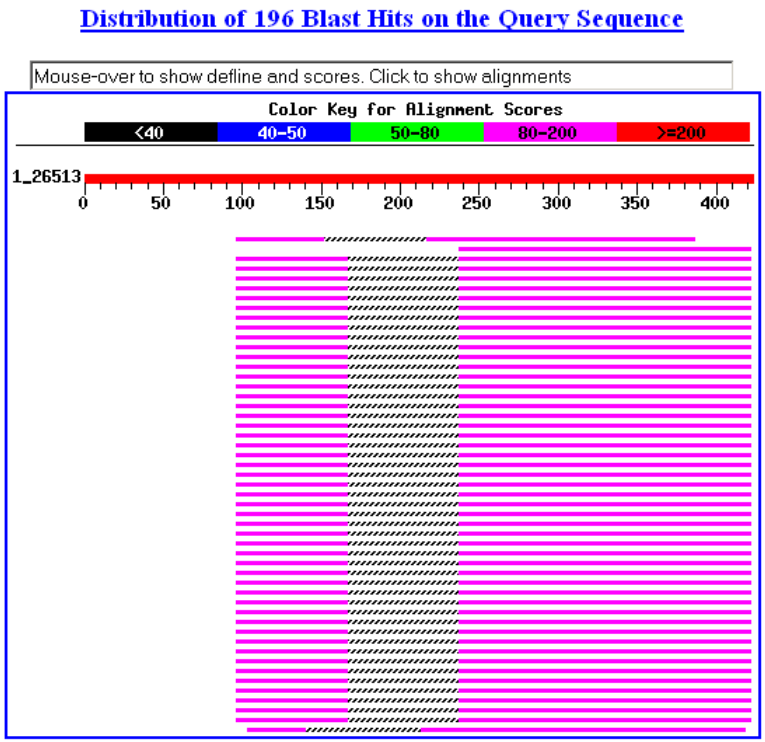
High sequence similarity usually implies significant functional or structural similarity and therefore conserved functions

Examples

- Searching databases for related sequences and sub sequences
- Comparing two or more sequences for similarities
- Reconstructing long sequences of DNA from overlapping sequence fragments (sequence assembly)
- Exploring frequently occurring patterns of nucleotides
- Finding informative elements in protein and DNA sequences
- Find differences between sequences (sequence variation)

Database searching

Heuristic - BLAST



Sequences producing significant alignments:		Score (bits)	E Value	
gi|28804810|dbj|AB104818.1|	Pipistrellus abramus HBA mRNA f...	115	1e-22	
gi|49420|emb|X57029.1|CAGLOBINM	M.auratus mRNA for alpha gl...	92	1e-15	
gi|40225899|gb|BC032122.2|	Homo sapiens hemoglobin, alpha 2...	84	3e-13	G U E
gi|30047787|gb|BC050661.1|	Homo sapiens hemoglobin, alpha 2...	84	3e-13	G U

Done

Adblock

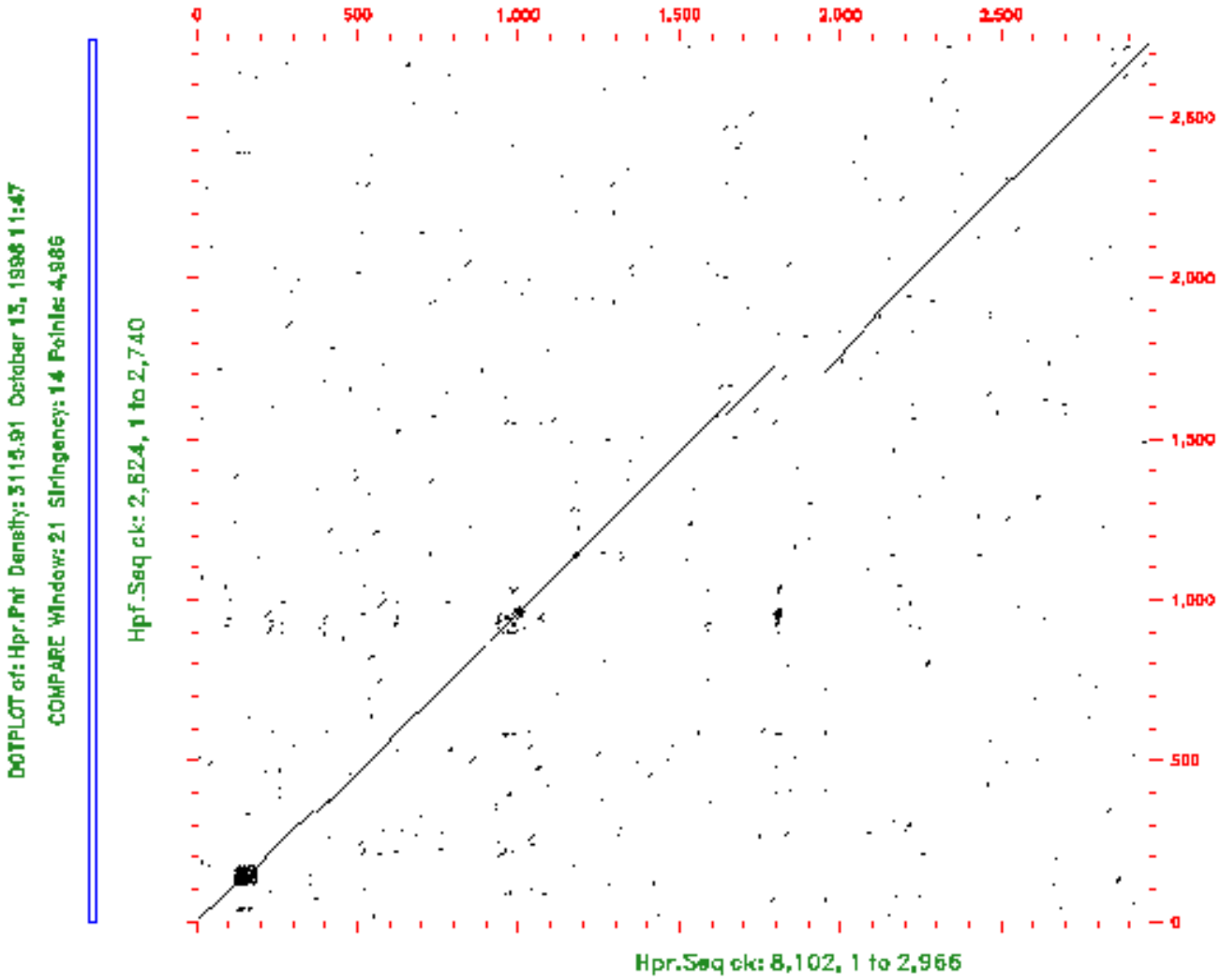
Database searching

Alignment

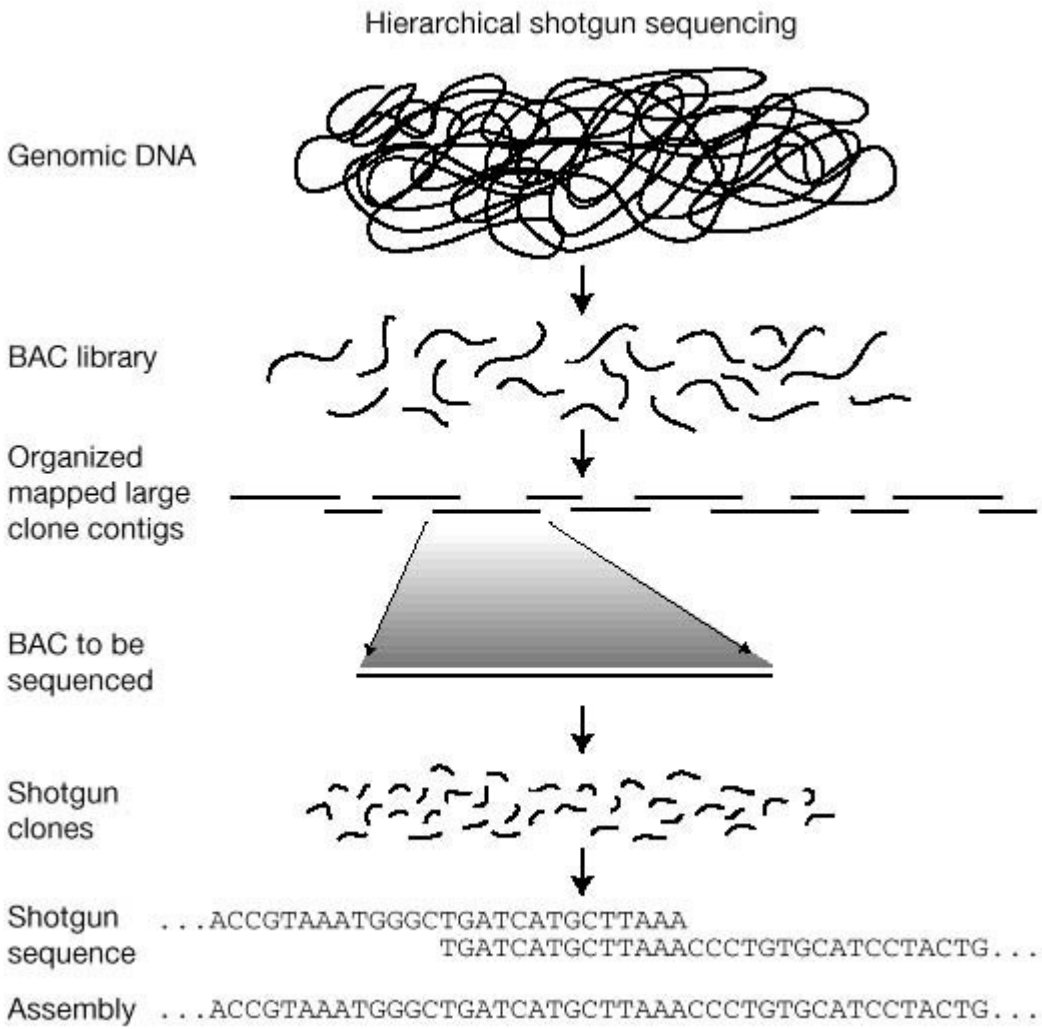
```
>gi|28804810|dbj|AB104818.1|   Pipistrellus abramus HBA mRNA for alpha globin, complete cds
          Length = 554

 Score =  115 bits (58), Expect = 1e-22
 Identities = 142/170 (83%)
 Strand = Plus / Plus


Query: 218 tggacgacctgccgtccgctctgtccgctctgtccgacctgcacgcttacaaactgcgtg 277
           |||||||||||||  |||| |||||||| |||  |||||||||||||  |||| |||||||
Sbjct: 248 tggacgacctgcccaccgccctgtccgccctgagcgacctgcacgcccacaagctgcgtg 307


Query: 278 ttgacccggttaacttcaaactgctgtcccactgcctgctggttaccctggctgctcacc 337
           | ||||| || |||||||| || |||  |||||||||||||||| |||||||||   ||||
Sbjct: 308 tggaccccgtcaacttcaagctcctgagccactgcctgctggtgaccctggcttgccacc 367


Query: 338 acccggctgaattcaccccggctgttcactccgacctggacaaattcctg 387
           |||| || |||||||||||||| || ||| ||  ||||||||| ||||||
Sbjct: 368 accccgccgaattcaccccggccgtgcacgcctccctggacaagttcctg 417




 Score = 58.0 bits (29), Expect = 2e-05
 Identities = 52/59 (88%), Gaps = 3/59 (5%)
 Strand = Plus / Plus


Query: 97  ttcctgtgcttcccgaccaccaaaacctacttcccgcactt---cgacctgtcccacgg 152
           |||||| ||||||| |||||||| |||||||||||| |||||   |||||||||||||||
Sbjct: 124 ttcctgagcttccccaccaccaagacctacttcccccacttctccgacctgtcccacgg 182




>gi|49420|emb|X57029.1|CAGLOBINM   M.auratus mRNA for alpha globin chain
          Length = 418

 Score = 91.7 bits (46), Expect = 1e-15
 Identities = 151/186 (81%)
```

Done Adblock

Comparing sequences
Dotplot



Reconstruct sequences
Sequence assembly

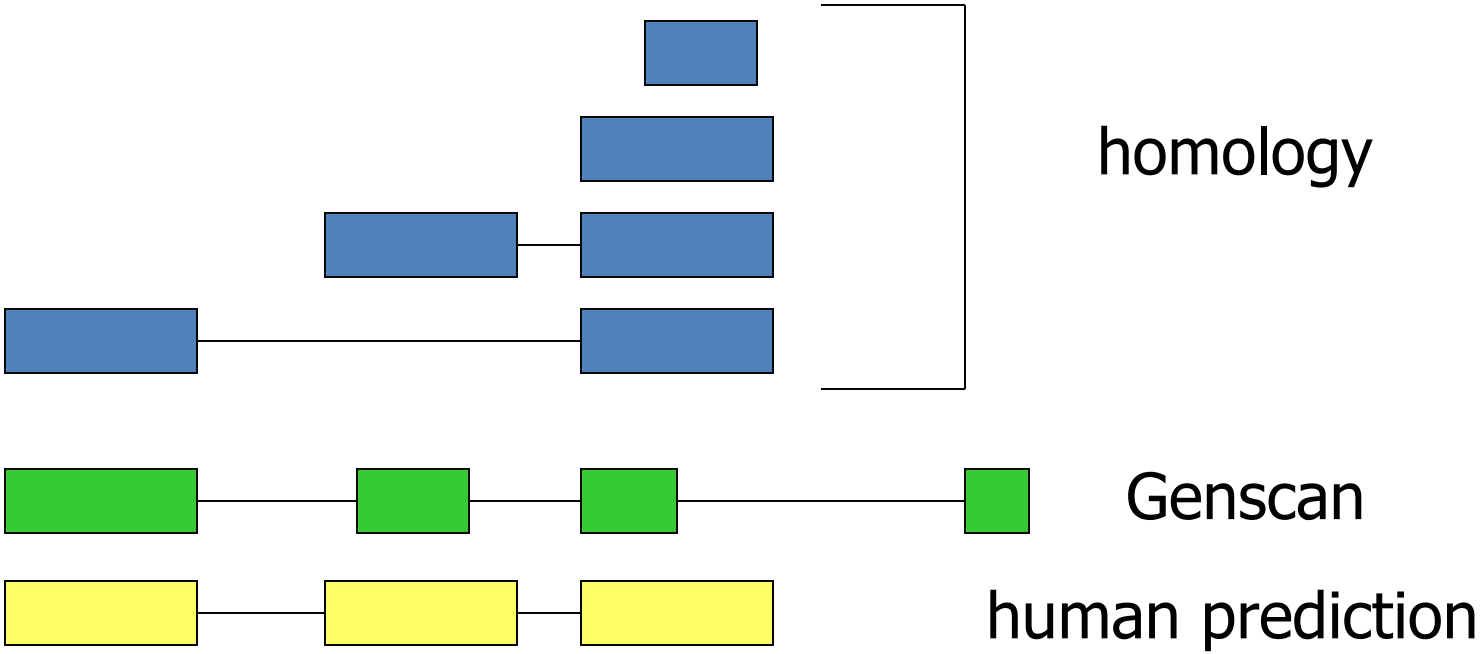


Lander et al (2001), Nature

Reconstructing long sequences of DNA from overlapping sequence fragments

Finding informative elements

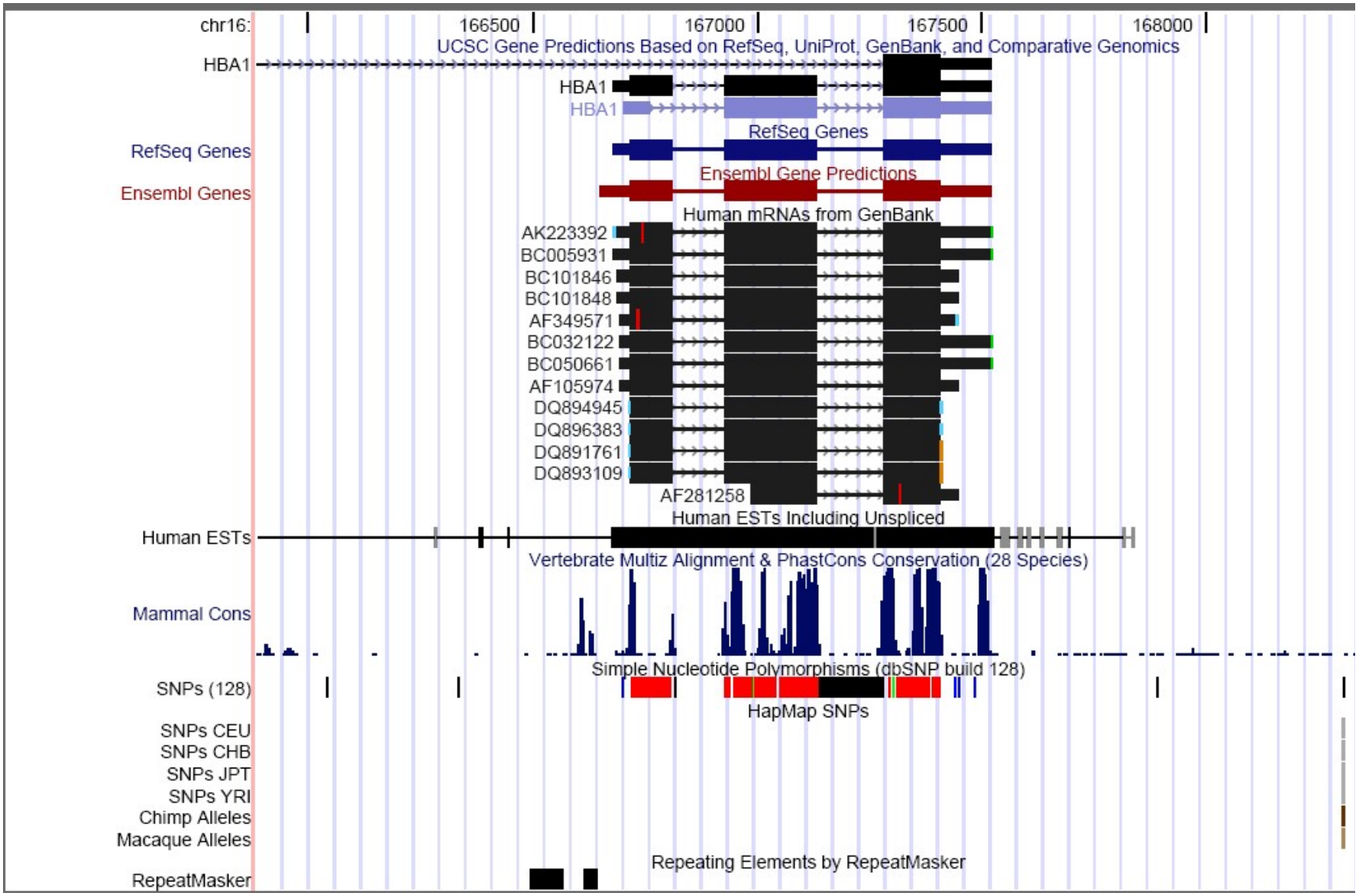
Automated gene annotation



IGI/IPI, OTTO, humans

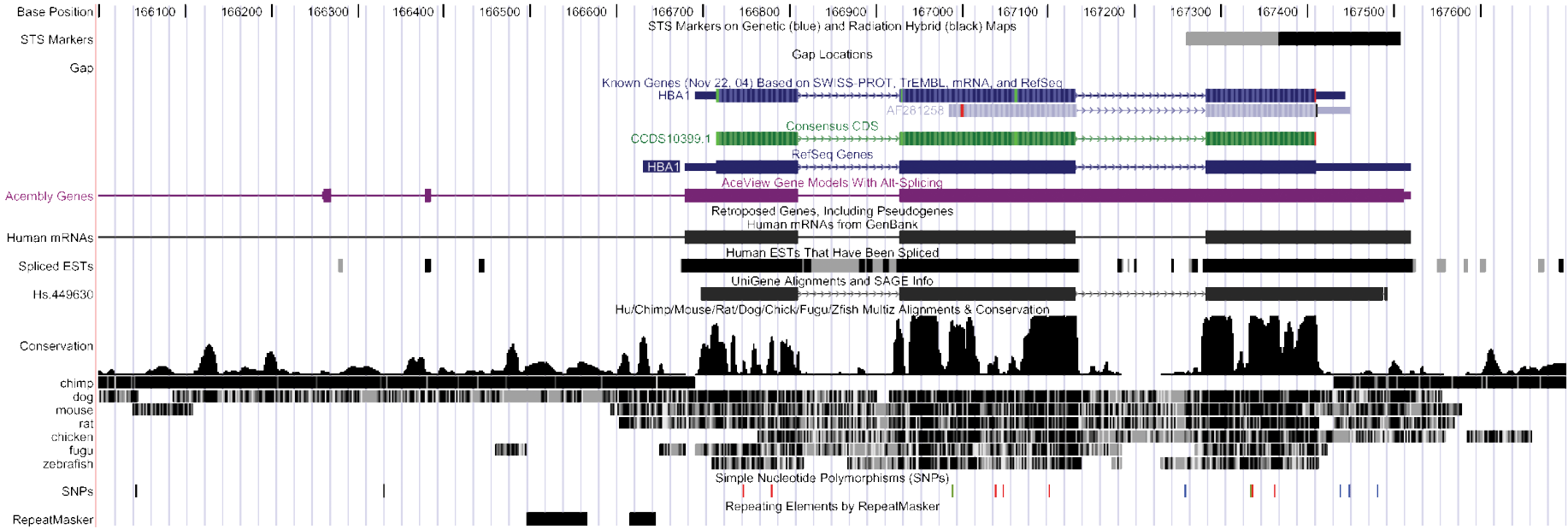
Finding informative elements

Gene annotation



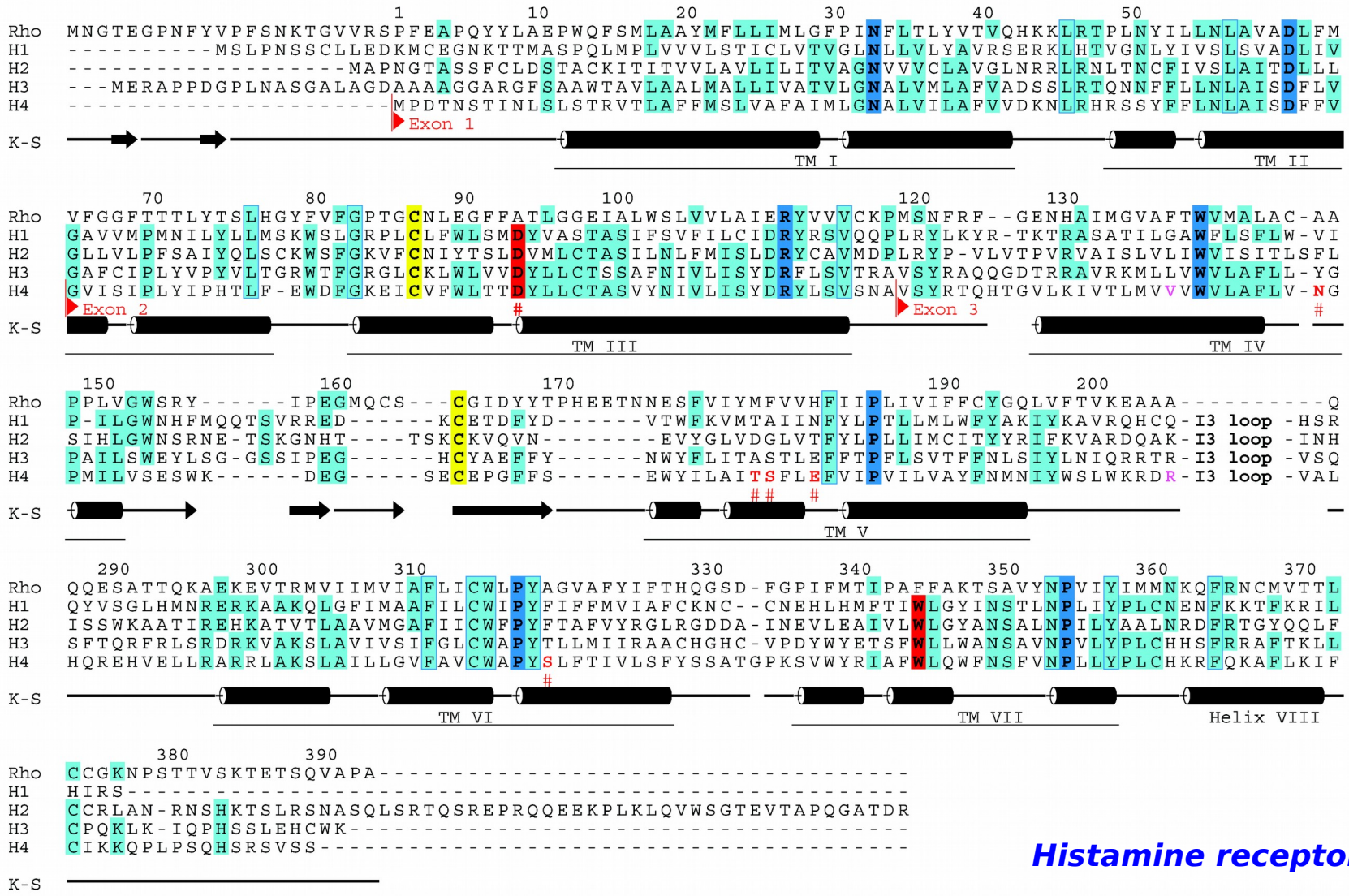
Finding informative elements

Conservation



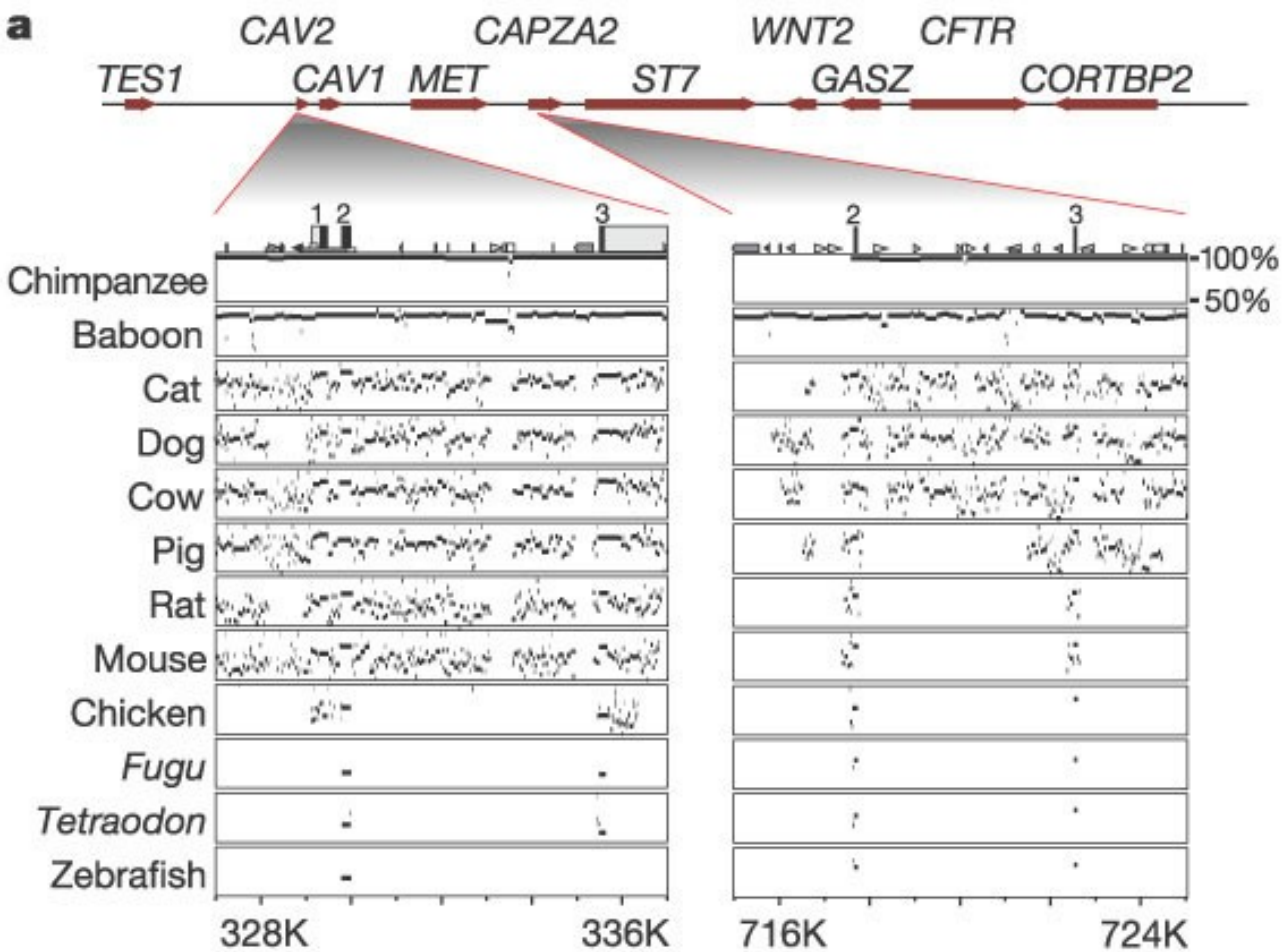
Finding informative elements

Multiple sequence alignment



Histamine receptors

Finding informative elements
Multiple sequence alignment



Thomas et al (2003), Nature

Sequence variation

Hemoglobin beta (HBB)

vs.

Hemoglobin beta in Sickle cell anemia (HBS)

```
Query 51 ATGGTGCATCTGACTCCTGAGGAGAAGTCTGCCGTTACTGCCCTGTGGGGCAAGGTGAAC 110
         |||||||| |||||||||| |||||||||| || |||||||| | ||||||||||||||||
Sbjct 1  ATGGTGCACCTGACTCCTGTGGAGAAGTCYGCNGTTACTGCNYTNTGGGGCAAGGTGAAC 60
```

Pairwise sequence comparison

Dotplot

Edit transcript

Sequence alignment

```
Sequen Sequence1 Sequence1 Sequence1 Sequence1
Sequence2 Seque2 Sequen   2Sequence2 Sequenc
```

Dotplot

Dot plots are two dimensional graphs, showing a comparison of two sequences

The two axis of the graph represent the two sequences being compared

Every region of the sequence is compared to every region of the other sequence

Dotplot

	T	C	G	C	A	G	T	C	C	C	C	G	G	C
C		x		x				x	x	x	x			x
A					x									
G			x			x						x	x	
C		x		x				x	x	x	x			x
A					x									
G			x			x						x	x	
C		x		x				x	x	x	x			x
A					x									
C		x		x				x	x	x	x			x
T	x						x							
C		x		x				x	x	x	x			x
A					x									
G			x			x						x	x	
C		x		x				x	x	x	x			x

Dotplot

Much noise without filter (four letter alphabet)

Search for a given number of matches (*stringency*) within a given range (*window*)

TCGCAGTCCCCGGC
CAGCAGCACTCAGC

Dotplot

Warning!

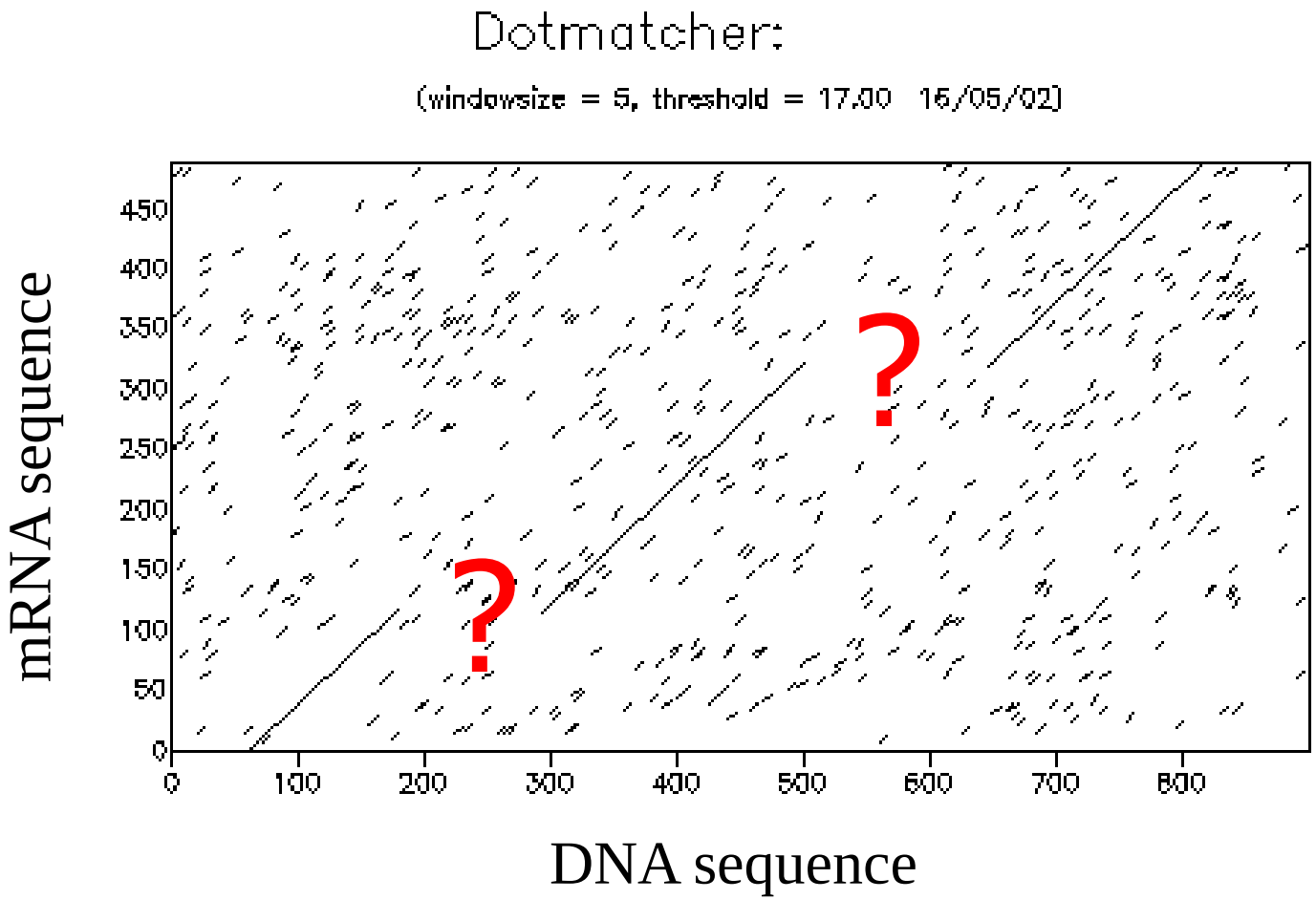
Window size smaller than region of similarity: segment pairs may have a score that is not sufficiently high enough

ABCDEFGH
XYZDXFGX

Window size to big: scores for unrelated amino acids may degrade the signal

ABCDEFGH
XYZDXFGX

Dotplot


Dotmatcher:
(windowsize = 5, threshold = 17.00 16/05/02)
mRNA sequence
DNA sequence
0
50
100
150
200
250
300
350
400
450
0
100
200
300
400
500
600
700
800
?
?

Edit transcript

Transformation of one string into other string

I – insertion

D – deletion

R – replacement (substitution)

M – match

Edit transcript

I – insertion
D – deletion
R – replacement
M – match

example:

S1 : vintner

S2 : writers

R	I	M	D	M	D	M	M	I
v		i	n	t	n	e	r	
w	r	i		t		e	r	s

Edit distance

Minimum number of operations
to transform S1 into S2

Edit transcript with minimal operations:
optimal transcript

Edit transcript *vs* Alignment

Edit transcript: shows mutation events

Distance is minimized

Alignment: display relationship between strings

Score is maximized

String alignment

S1 = `qacdbd`

S2 = `qawxb`

q	a	c	-	d	b	d
q	a	w	x	-	b	-

String alignment - competition

Sequence alignment

- Introduction sequence alignment
- **Calculation of an alignment**
  - **exercises**
- Different types of the algorithm
  - exercises
- Wrap-up

Dynamic programming

Dynamic programming

3 components:

1. Recurrence relation (*initialization*)
2. Tabular computation (*matrix filling*)
3. Traceback

Global alignment

Input:

Two sequences S1 and S2 of roughly the same length

S1 = **acbcdb** S2 = **cadbd**

Question:

What is the best similarity between them? Find best alignment

a	c	-	-	b	c	d	b
-	c	a	d	b	-	d	-

Some definitions

Two sequences: *S1[1..i]* and *S2[1..j]*
with lengths: *n* and *m*

S1:

1	2	3	4	5	6
a	c	b	c	d	b

S2:

1	2	3	4	5
c	a	d	b	d

Each character of *S1* (i = 0..n)
is compared with *S2* (j = 0..m)

i = 0, j=0: "-"

Recurrences global alignment

Base: $V(i,0) = \sum_{1 \le k \le i} s(S1(k), -)$

$V(0,j) = \sum_{1 \le k \le j} s(-, S2(k))$

Score from substitution matrix

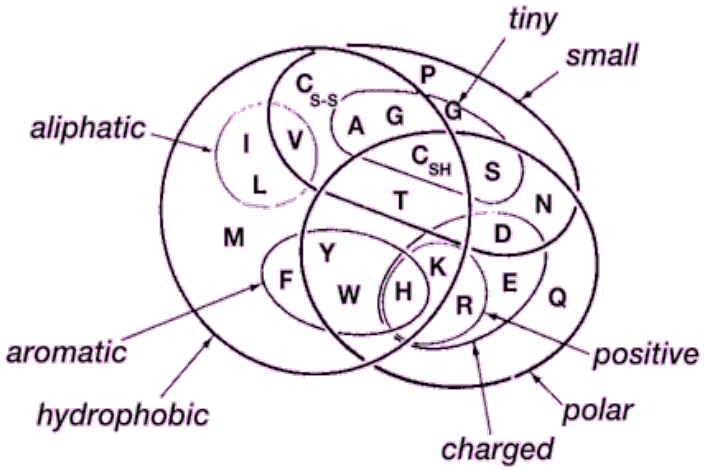
For i>0 and j>0

$$
\begin{aligned}
V(i,j) = \max [& \\
& V(i-1, j-1) + s(S1(i), S2(j)), \\
& V(i-1, j) + s(S1(i), -), \\
& V(i, j-1) + s(-, S2(j)) \\
] &
\end{aligned}
$$

Substitution scores

Scoring matrix based on:

- genetic code
- physical-chemical properties
- studies of molecular structure
- studies of evolution
- statistics
- …



BLOSUM100 Scoring Matrix

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	8	-3	-4	-5	-2	-2	-3	-1	-4	-4	-4	-2	-3	-5	-2	1	-1	-6	-5	-2	-4	-2	-2	-10
R		10	-2	-5	-8	0	-2	-6	-1	-7	-6	3	-4	-6	-5	-3	-3	-7	-5	-6	-4	-1	-3	-10
N			11	1	-5	-1	-2	-2	0	-7	-7	-1	-5	-7	-5	0	-1	-8	-5	-7	5	-2	-3	-10
D				10	-8	-2	2	-4	-3	-8	-8	-3	-8	-8	-5	-2	-4	-10	-7	-8	6	0	-4	-10
C					14	-7	-9	-7	-8	-3	-5	-8	-4	-4	-8	-3	-3	-7	-6	-3	-7	-8	-5	-10
Q						11	2	-5	1	-6	-5	2	-2	-6	-4	-2	-3	-5	-4	-5	-2	5	-2	-10
E							10	-6	-2	-7	-7	0	-5	-8	-4	-2	-3	-8	-7	-5	0	7	-3	-10
G								9	-6	-9	-8	-5	-7	-8	-6	-2	-5	-7	-8	-8	-3	-5	-4	-10
H									13	-7	-6	-3	-5	-4	-5	-3	-4	-5	1	-7	-2	-1	-4	-10
I										8	2	-6	1	-2	-7	-5	-3	-6	-4	4	-8	-7	-3	-10
L											8	-6	3	0	-7	-6	-4	-5	-4	0	-8	-6	-3	-10
K												10	-4	-6	-3	-2	-3	-8	-5	-5	-2	0	-3	-10
M													12	-1	-5	-4	-2	-4	-5	0	-7	-4	-3	-10
F														11	-7	-5	-5	0	4	-3	-7	-7	-4	-10
P															12	-3	-4	-8	-7	-6	-5	-4	-4	-10
S																9	2	-7	-5	-4	-1	-2	-2	-10
T																	9	-7	-5	-1	-2	-3	-2	-10
W																		17	2	-5	-9	-7	-6	-10
Y																			12	-5	-6	-6	-4	-10
V																				8	-7	-5	-3	-10
B																					6	0	-4	-10
Z																						6	-2	-10
X																							-3	-10
*																								1

Substitution scores

Common in protein alignments
Also suggested for nucleotide sequences:

It is more likely that an A is replaced by a G than a T

Scoring DNA sequences:
Most of the time identities are scored
instead of putting weight to substitutions

Tabular computation

S2 j = 0 .. m

0 1 2 3 4 5

S1 i = 0 .. n

		-	c	a	d	b	d
0	-						
1	a						
2	c						
3	b						
4	c						
5	d						
6	b						

Fill in base conditions

Tabular computation

S2 ***(j)***

S1 ***(i)***

	-	c	a	d	b	d
-	0	-1	-2	-3	-4	-5
a	-1					
c	-2					
b	-3					
c	-4					
d	-5					
b	-6					

$$V(i,0) = \sum_{1 \le k \le i} s(S1(k), -)$$

$$V(0,j) = \sum_{1 \le k \le j} s(-, S2(k))$$

Scores:
match = 2
mismatch = -1
space = -1

Tabular computation

V(i-1,j-1) *V(i-1,j)* *S2*

V(i,j-1)

S1

	-	c	a	d	b	d
-	0	-1	-2	-3	-4	-5
a	-1	*V(i,j)*				
c	-2					
b	-3					
c						
d						
b						

V(i,j) = max [
V(i-1, j-1) + s(S1(i), S2(j)),
V(i-1, j) + s(S1(i), -),
V(i, j-1) + s(-, S2(j))
]

Scores:
match = 2
mismatch = -1
space = -1

Tabular computation

S2

S1

	-	c	a	d	b	d
-	0	-1	-2	-3	-4	-5
a	-1	-1				
c	-2					
b	-3					
c						
d						
b						

$$V(i,j) = \max [\; V(i-1, j-1) + s(S1(i), S2(j)),\; V(i-1, j) + s(S1(i), -),\; V(i, j-1) + s(-, S2(j)) \;]$$

Scores:
match = 2
mismatch = -1
space = -1

Tabular computation

S2

S1

	-	c	a	d	b	d
-	0	-1	-2	-3	-4	-5
a	-1	-1	1			
c	-2					
b	-3					
c						
d						
b						

$$V(i,j) = \max [\; V(i-1, j-1) + s(S1(i), S2(j)),\; V(i-1, j) + s(S1(i), -),\; V(i, j-1) + s(-, S2(j)) \;]$$

Scores:
match = 2
mismatch = -1
space = -1

Tabular computation

S2

S1

	-	c	a	d	b	d
-	0	-1	-2	-3	-4	-5
a	-1	-1	1	0		
c	-2					
b	-3					
c						
d						
b						

$$V(i,j) = \max [\; V(i-1, j-1) + s(S1(i), S2(j)),\; V(i-1, j) + s(S1(i), -),\; V(i, j-1) + s(-, S2(j)) \;]$$

Scores:
match = 2
mismatch = -1
space = -1

Tabular computation

S2

S1

	-	c	a	d	b	d
-	0	-1	-2	-3	-4	-5
a	-1	-1	1	0	-1	-2
c	-2	1				
b	-3					
c						
d						
b						

$$V(i,j) = \max [\; V(i-1, j-1) + s(S1(i), S2(j)),\; V(i-1, j) + s(S1(i), -),\; V(i, j-1) + s(-, S2(j)) \;]$$

Scores:
match = 2
mismatch = -1
space = -1

Tabular computation

S2

S1

	-	c	a	d	b	d
-	0	-1	-2	-3	-4	-5
a	-1	-1	1	0	-1	-2
c	-2	1	0			
b	-3					
c						
d						
b						

$$V(i,j) = \max [\; V(i-1, j-1) + s(S1(i), S2(j)),\; V(i-1, j) + s(S1(i), -),\; V(i, j-1) + s(-, S2(j)) \;]$$

Scores:
match = 2
mismatch = -1
space = -1

Tabular computation

Scores:
match = 2
mismatch = -1
space = -1

S2

S1

	-	c	a	d	b	d
-	0	-1	-2	-3	-4	-5
a	-1	-1	1	0	-1	-2
c	-2	1	0	0	-1	-2
b	-3	0	0	-1	2	1
c	-4	-1	-1	-1	1	1
d	-5	-2	-2	1	0	3
b	-6	-3	-3	0	3	2

Tabular computation

S2

S1

	-	c	a	d	b	d
-	0	-1	-2	-3	-4	-5
a	-1	↖ -1	↖ 1	← 0	← -1	← -2
c	-2	↖ 1	↑← 0	↖ 0	↖← -1	↖← -2

Three possible moves in the *traceback:*

- *Diagonal*: letters are aligned
- *Left*: a gap is introduced in the left sequence (S1)
- *Up*: a gap in the top sequence (S2)

Trace back

- a c b c d b
c a d b - d -

S2

S1

	-	c	a	d	b	d
-	0	-1	-2	-3	-4	-5
a	-1	-1	1	0	-1	-2
c	-2	1	0	0	-1	-2
b	-3	0	0	-1	2	1
c	-4	-1	-1	-1	1	1
d	-5	-2	-2	1	0	3
b	-6	-3	-3	0	3	2

Calculate score

Scores:
match = 2
mismatch = -1
space = -1

```
a c b c d b -
- c - a d b d

a c b c d b -
- c a - d b d

- a c b c d b
c a d b - d -
```

matches: 3

mismatches: 1

spaces: 3

(2*3) + (-1*1) + (-1*3)

= 2

Recurrences global alignment

Base: $V(i,0) = \Sigma\, s(S1(k), -)$ $\quad 1 \le k \le i$

$V(0,j) = \Sigma\, s(-, S2(k))$ $\quad 1 \le k \le j$

For i>0 and j>0

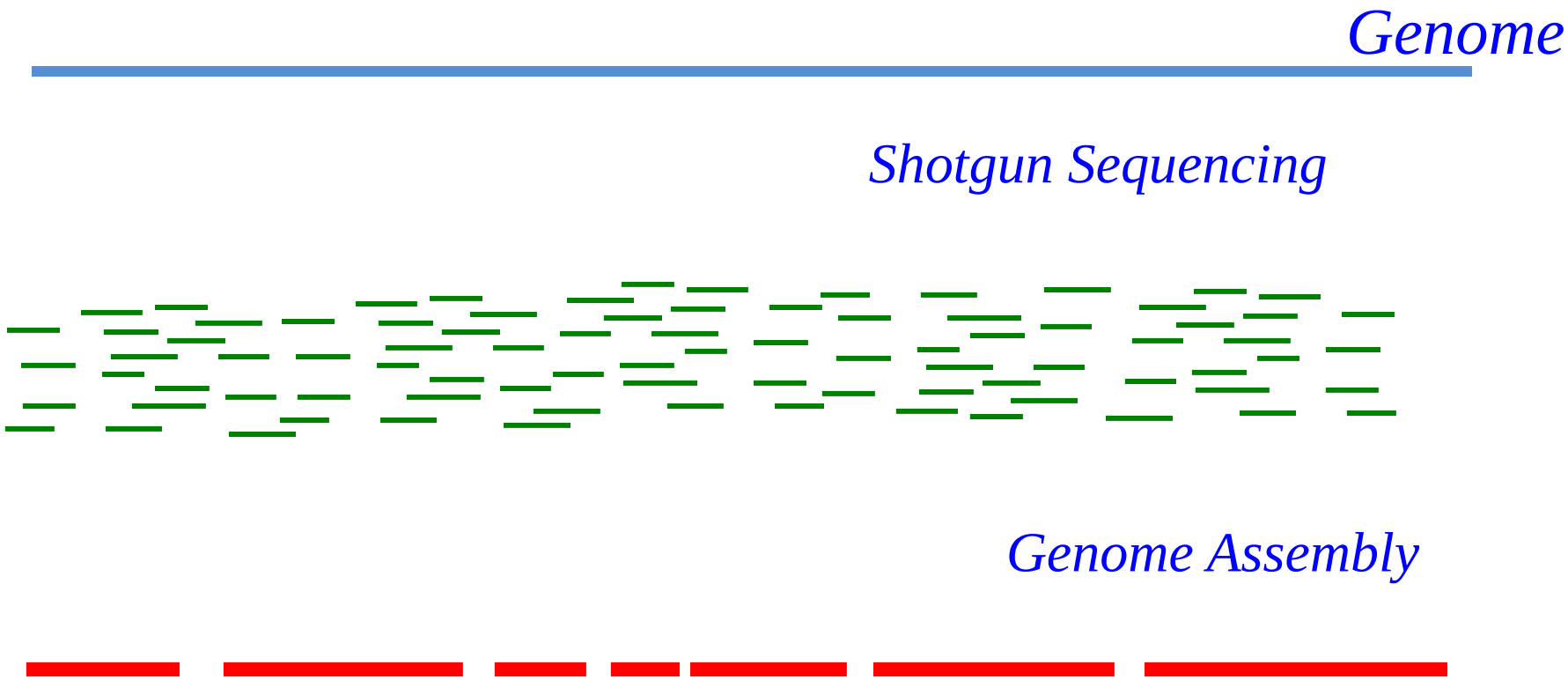
$$
\begin{aligned}
V(i,j) = \max [& \\
& V(i-1, j-1) + s(S1(i), S2(j)), \\
& V(i-1, j) + s(S1(i), -), \\
& V(i, j-1) + s(-, S2(j)) \\
] &
\end{aligned}
$$

Sequence alignment

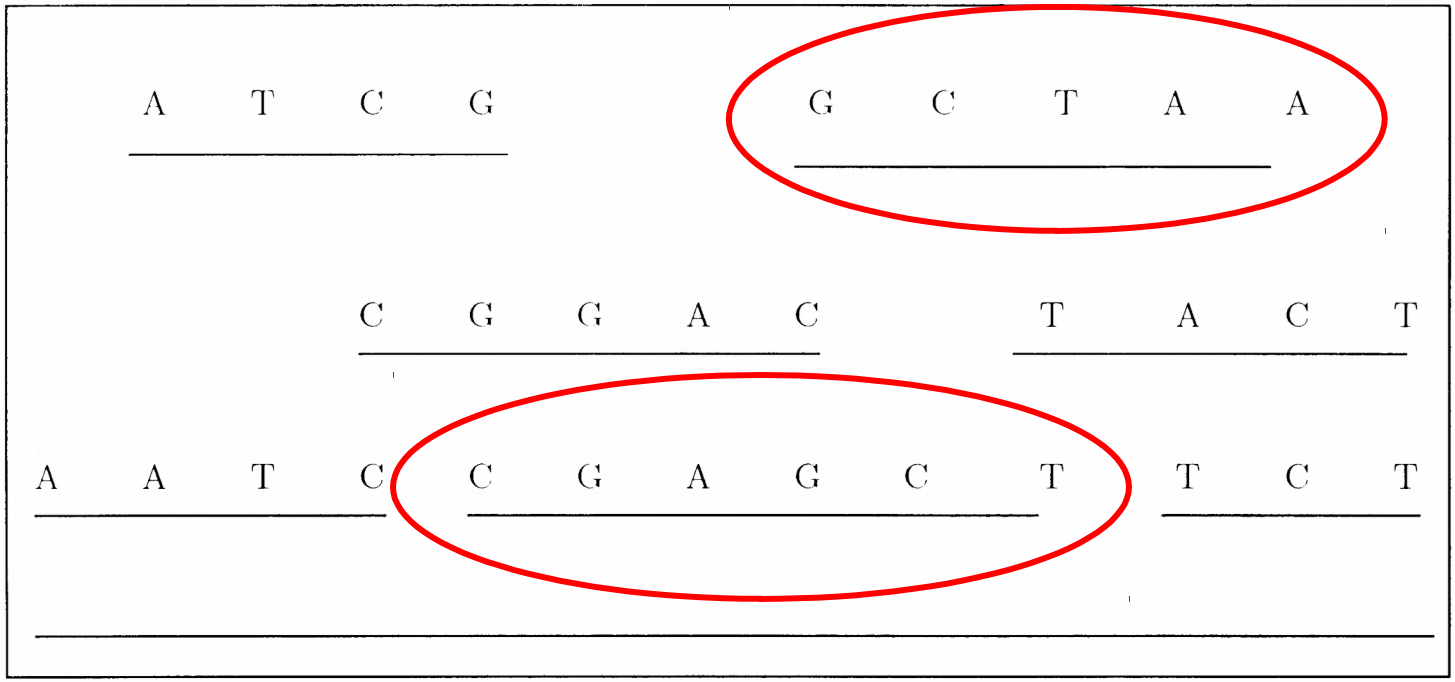
- Introduction sequence alignment
- Calculation of an alignment
  - exercises
- **Different types of the algorithm**
  - **exercises**
- Wrap-up

Ends-free alignment

Motivation: Shotgun sequence assembly



Ends-free alignment



Recurrences ends-free alignment

Base: $V(i,0) = 0$

$V(0,j) = 0$

For i>0 and j>0

$V(i,j) = \max [$

$V(i-1, j-1) + s(S1(i), S2(j)),$

$V(i-1, j) + s(S1(i), -),$

$V(i, j-1) + s(-, S2(j))$

$]$

Tabular computation

S2

S1

	-	g	c	t	a	a
-	0	0	0	0	0	0
c	0					
g	0					
a	0					
g	0					
c	0					
t	0					

$V(i,0) = 0$

$V(0,j) = 0$

~~Scores:~~
~~match = 2~~
~~mismatch = -1~~
~~space = -1~~

Tabular computation

S2

S1

	-	g	c	t	a	a
-	0	0	0	0	0	0
c	0	-1				
g	0					
a	0					
g						
c						
t						

$$V(i,j) = \max [\; V(i-1, j-1) + s(S1(i), S2(j)),\; V(i-1, j) + s(S1(i), -),\; V(i, j-1) + s(-, S2(j)) \;]$$

Scores:
match = 2
mismatch = -1
space = -1

Tabular computation

Scores:
match = 2
mismatch = -1
space = -1

S2

S1

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

Trace back

S2

S1

	-	g	c	t	a	a
-	0	0	0	0	0	0
c	0	-1	2	1	0	-1
g	0	[illegible]	[illegible]	1	0	-1
a	0	[illegible]	[illegible]	[illegible]	3	2
g	0	2	1	0	2	2
c	0	1	4	3	2	1
t	0	0	3	6	5	4

Ends are free
find highest value
in row n or col m

Trace back

```
cga g c t --
--- g c t aa
```

S2

S1

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

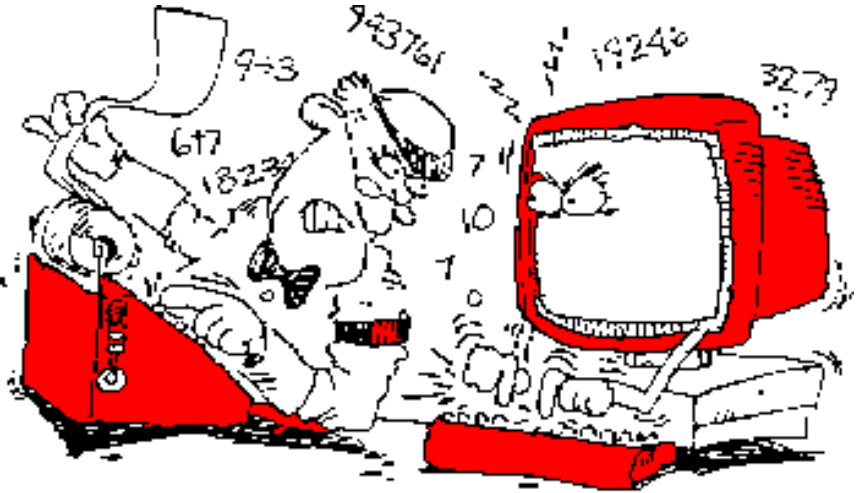
Recurrences ends-free alignment

Base: $V(i,0) = 0$

$V(0,j) = 0$

For i>0 and j>0

$$V(i,j) = \max [\; V(i-1, j-1) + s(S1(i), S2(j)), \; V(i-1, j) + s(S1(i), -), \; V(i, j-1) + s(-, S2(j)) \;]$$



Local alignment

Finds regions of sequence with a high degree of similarity

Better at finding motifs, especially for sequences that are different overall

Will return only the best matching segment for a given pair of sequences

Global vs Local

Global : *Needleman-Wunsch*

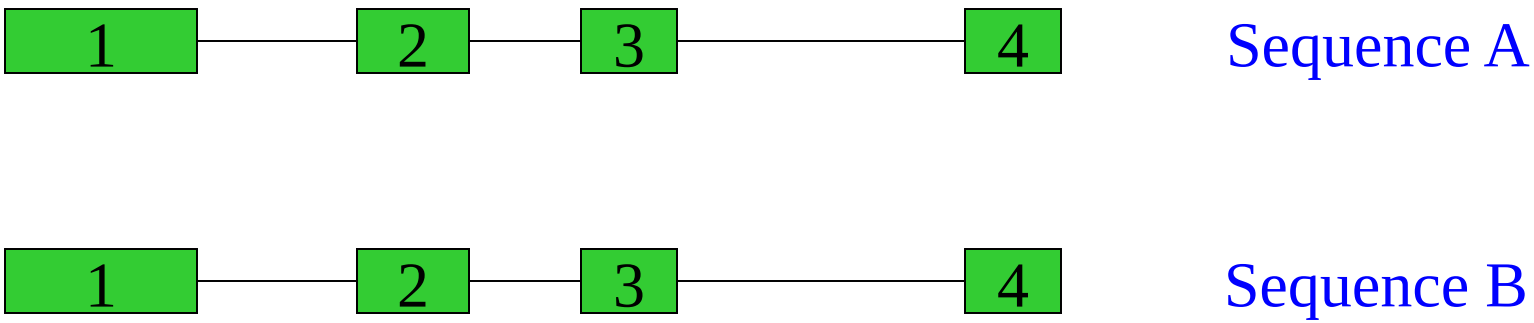
Local : *Smith-Waterman*

Local alignment far more meaningful in many biological applications

In some situations global alignment seems to be more effective in exposing important biological commonalities

Global vs Local

Example



Local alignment

Definition: Two sequences S1 and S2

Question: *Find subsequences α and β of S1 and S2 whose similarity is maximum over all such pairs of subsequence*

example:

S1 = g g t c t g a g

S2 = a a a c g a

Subseq α = c t g a

Subseq β = c - g a

Recurrences local alignment

Base: $V(i,0) = 0$

$V(0,j) = 0$

For i>0 and j>0

$$V(i,j) = \max [\; 0, \; V(i-1, j-1) + s(S1(i), S2(j)), \; V(i-1, j) + s(S1(i), -), \; V(i, j-1) + s(-, S2(j)) \;]$$

Tabular computation

S2

S1

	-	x	x	c	d	e	f
-	0	0	0	0	0	0	0
a	0						
b	0						
c	0						
x	0						
d	0						
e	0						
x	0						

$V(i,0) = 0$

$V(0,j) = 0$

Tabular computation

S2

S1

	-	x	x	c	d	e	f
-	0	0	0	0	0	0	0
a	0	0					
b	0						
c	0						
x							
d							
e							
x							

$$V(i,j) = \max [\ 0,\ V(i-1, j-1) + s(S1(i), S2(j)),\ V(i-1, j) + s(S1(i), -),\ V(i, j-1) + s(-, S2(j))\ ]$$

Scores:
match = 2
mismatch = -1
space = -1

Tabular computation

S2

S1

	-	x	x	c	d	e	f
-	0	0	0	0	0	0	0
a	0	0	0	0	0	0	0
b	0	0	0	0	0	0	0
c	0	0	0				
x							
d							
e							
x							

$$V(i,j) = \max [\ 0,\ V(i-1, j-1) + s(S1(i), S2(j)),\ V(i-1, j) + s(S1(i), -),\ V(i, j-1) + s(-, S2(j))\ ]$$

Scores:
match = 2
mismatch = -1
space = -1

Tabular computation

S2

S1

	-	x	x	c	d	e	f
-	0	0	0	0	0	0	0
a	0	0	0	0	0	0	0
b	0	0	0	0	0	0	0
c	0	0	0	2			
x							
d							
e							
x							

$$V(i,j) = \max [\ 0,\ V(i-1, j-1) + s(S1(i), S2(j)),\ V(i-1, j) + s(S1(i), -),\ V(i, j-1) + s(-, S2(j))\ ]$$

Scores:
match = 2
mismatch = -1
space = -1

Tabular computation

S2

S1

	-	x	x	c	d	e	f
-	0	0	0	0	0	0	0
a	0	0	0	0	0	0	0
b	0	0	0	0	0	0	0
c	0	0	0	2	1		
x							
d							
e							
x							

$$V(i,j) = \max [\ 0,\ V(i-1, j-1) + s(S1(i), S2(j)),\ V(i-1, j) + s(S1(i), -),\ V(i, j-1) + s(-, S2(j))\ ]$$

Scores:
match = 2
mismatch = -1
space = -1

Tabular computation

S2

S1

	-	x	x	c	d	e	f
-	0	0	0	0	0	0	0
a	0	0	0	0	0	0	0
b	0	0	0	0	0	0	0
c	0	0	0	2	1	0	0
x							
d							
e							
x							

$$V(i,j) = \max [\ 0,\ V(i-1, j-1) + s(S1(i), S2(j)),\ V(i-1, j) + s(S1(i), -),\ V(i, j-1) + s(-, S2(j))\ ]$$

Scores:
match = 2
mismatch = -1
space = -1

Tabular computation

Scores:
match = 2
mismatch = -1
space = -1

S2

S1

	-	x	x	c	d	e	f
-	0	0	0	0	0	0	0
a	0	0	0	0	0	0	0
b	0	0	0	0	0	0	0
c	0	0	0	2	1	0	0
x	0	2	2	1	1	0	0
d	0	1	1	1	3	2	1
e	0	0	0	0	2	5	4
x	0	2	2	1	1	4	4

Trace back

c x d e
c - d e

S2

S1

Find largest value in any cell
Trace back until cell(i,j) is

	-	x	x	c	d	e	f
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
b	0	0	0	0	0	0	0
c	0	0	0	2	1	0	0
x	0	2	2	1	1	0	0
d	0	1	1	1	3	2	1
e	0	0	0	0	2	5	4
x	0	2	2	1	1	4	4

Calculate score

Scores:
match = 2
mismatch = -1
space = -1

c x d e
c - d e

x - d e
x c d e

matches: 3
mismatches: 0
spaces: 1
(2*3) + (-1*0) + (-1*1)
= 5

Recurrences local alignment

Base: $V(i,0) = 0$

$V(0,j) = 0$

For i>0 and j>0

$$V(i,j) = \max [\; 0, \; V(i-1, j-1) + s(S1(i), S2(j)), \; V(i-1, j) + s(S1(i), -), \; V(i, j-1) + s(-, S2(j)) \;]$$

Gapped alignments

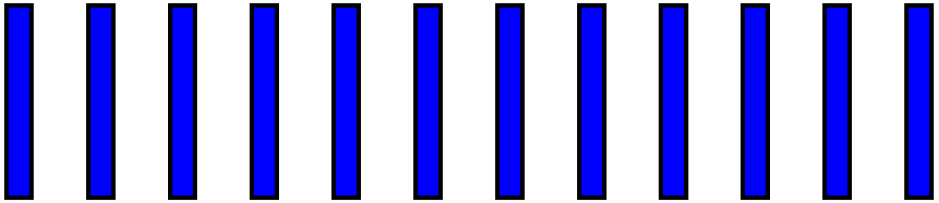
Theory: single mutation can delete or insert several nucleotides or amino acids

Insertion of a gap must improve the quality of the alignment (raise quality score)

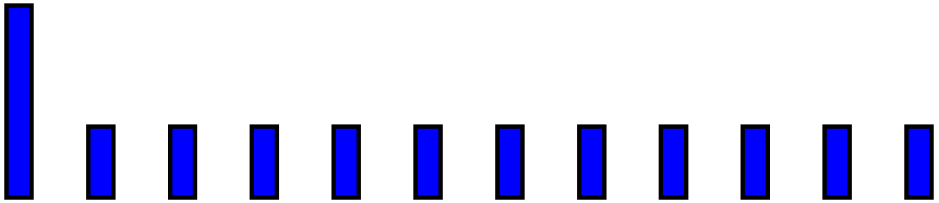
Different types of models

Gapped alignments

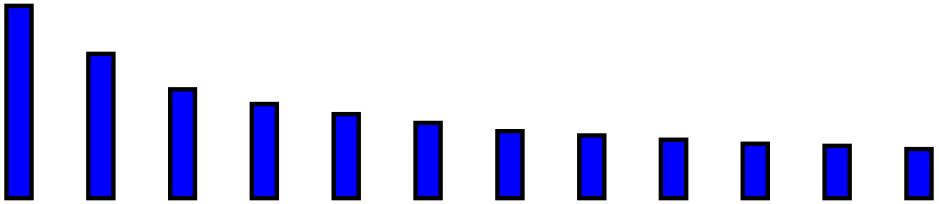
Constant



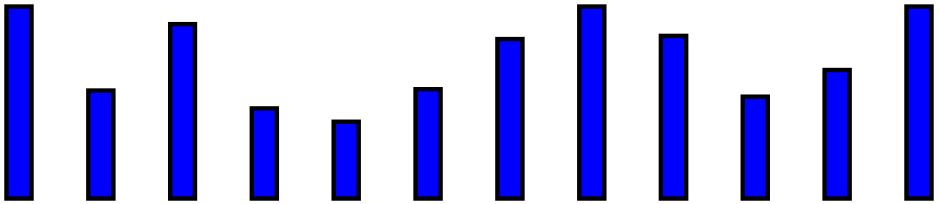
Affine



Convex



Arbitrary



Gapped alignments

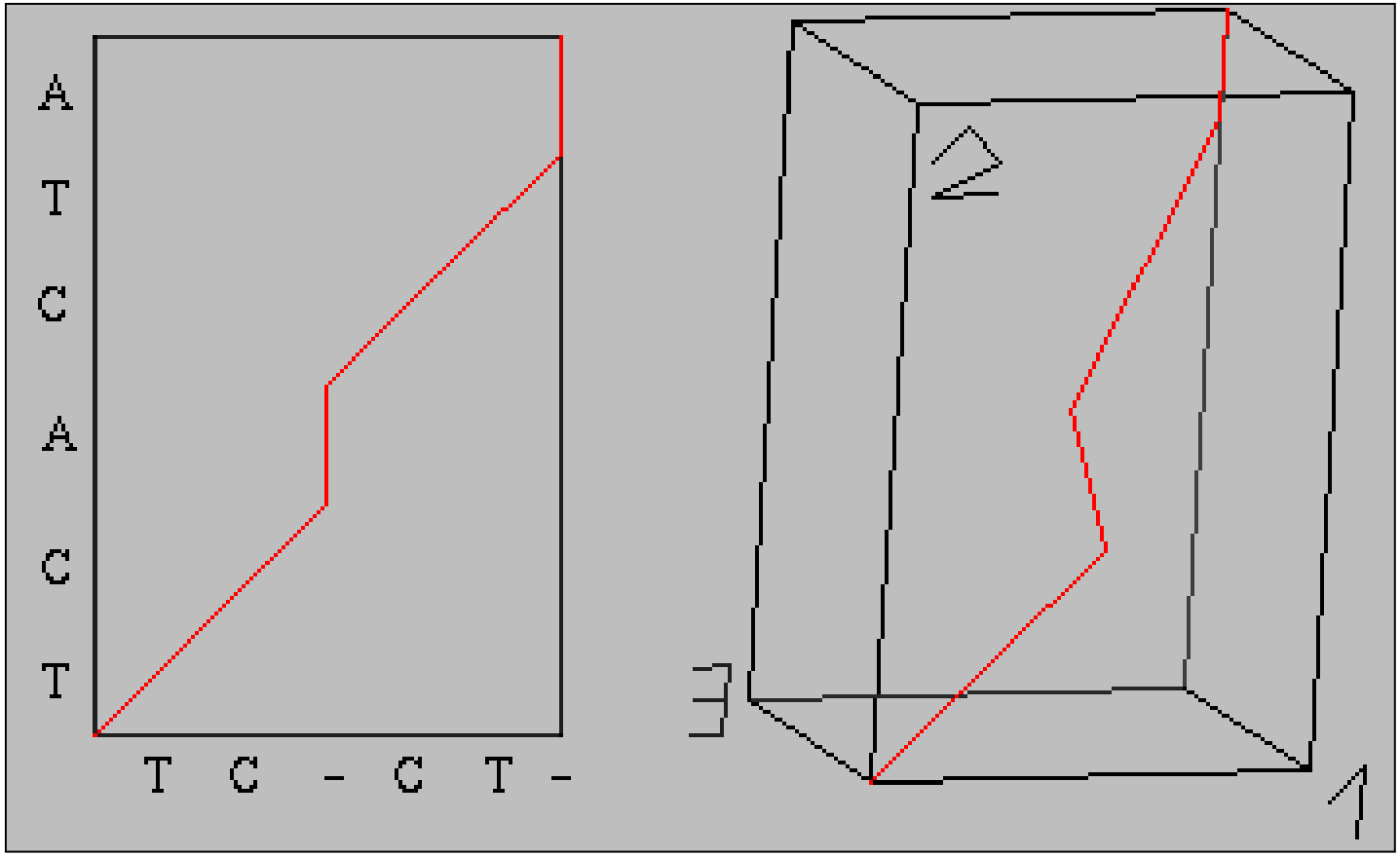
Affine is most used model

Gap opening/creation penalty (GOP): high

Gap extending penalty (GEP): low

Multiple alignment

Multiple alignment



http://bibiserv.techfak.uni-bielefeld.de/visualign/

Motivation

- It may reveal evolutionary commonalities
  - Conserved motifs
  - Common 2 and 3-dimensional structure
  - Clues about common biological function

- It is used for characterizing families or superfamilies of proteins

Algorithms

Dynamic programming in higher dimensions

+ true optimization

- execution time & memory requirements grow exponentially

Algorithms

Perform all pairwise alignments

Create consensus of two most similar pair

Add more sequences to this generalized – *consensus* - sequence

+ fast

- errors in early alignments affect rest of alignments

Correct mistakes

Remove sequences and realign them

Find short highly conserved regions, align them and repeat procedure on unaligned regions

Sequence alignment

- Introduction sequence alignment
- Calculation of an alignment
  - exercises
- Different types of the algorithm
  - exercises
- **Wrap-up**

Algorithms for computing sequence alignments

Global

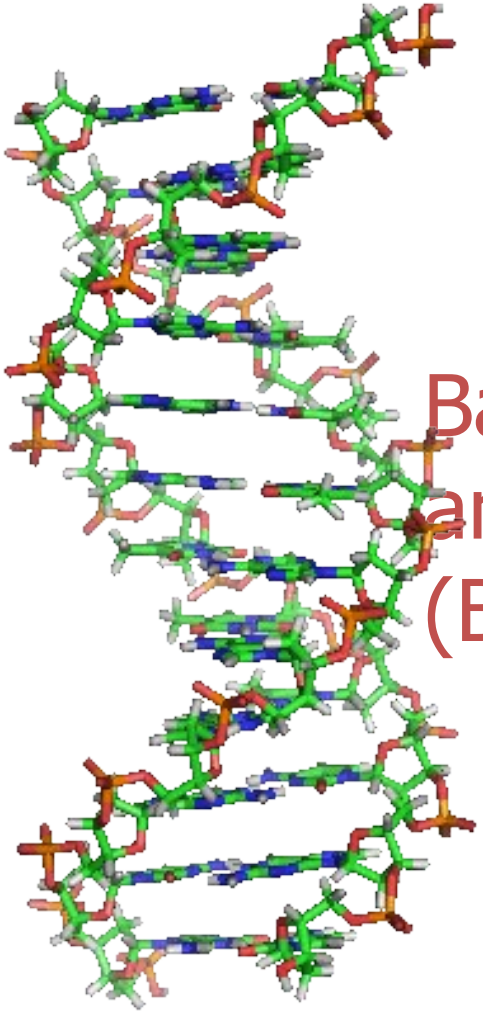
find overall alignment, rarely used

Ends-free

find overlap in sequence ends, sequence assembly

Local

find similar sub-sequences, often used



Basic Local Alignment and Search Tool (BLAST)