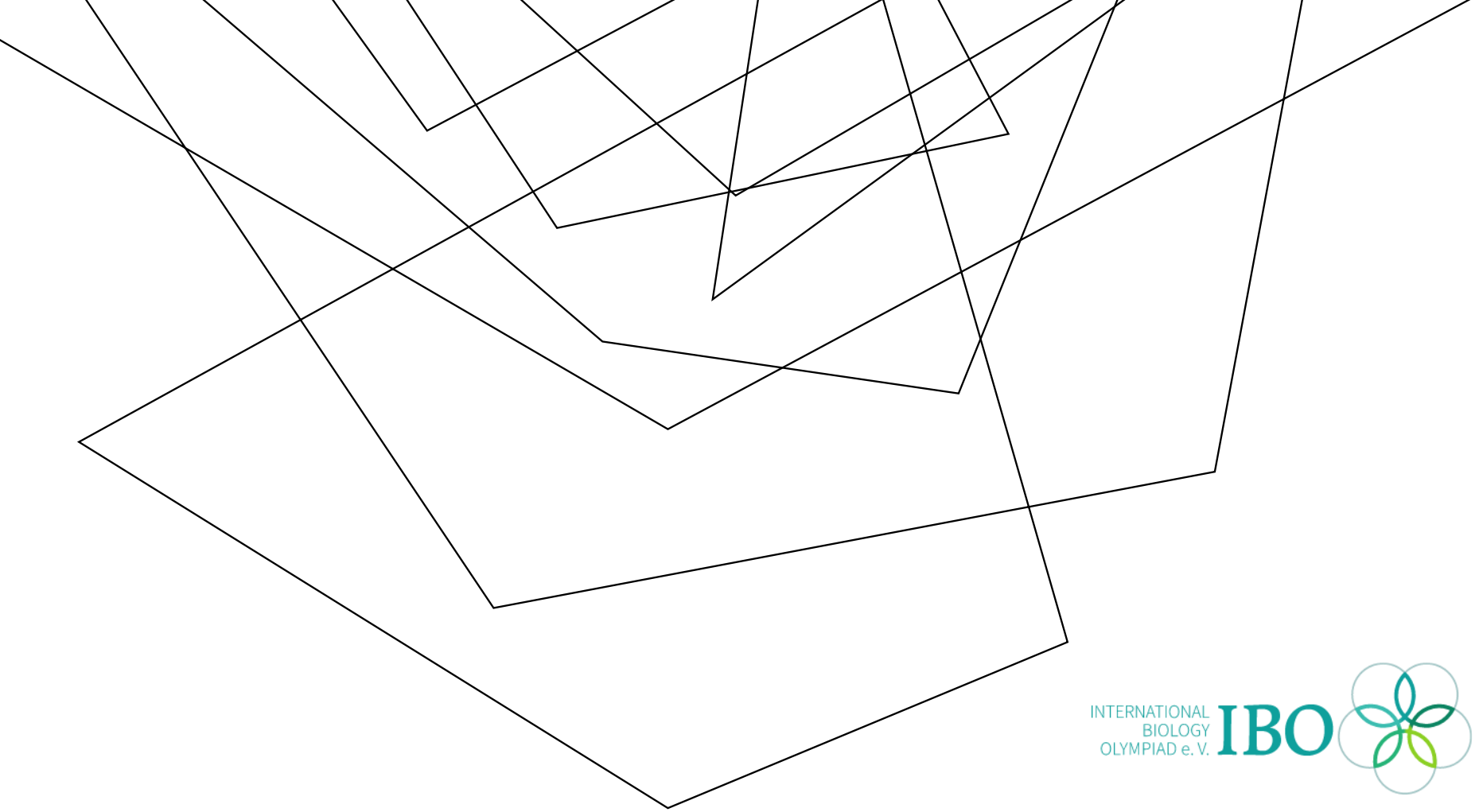
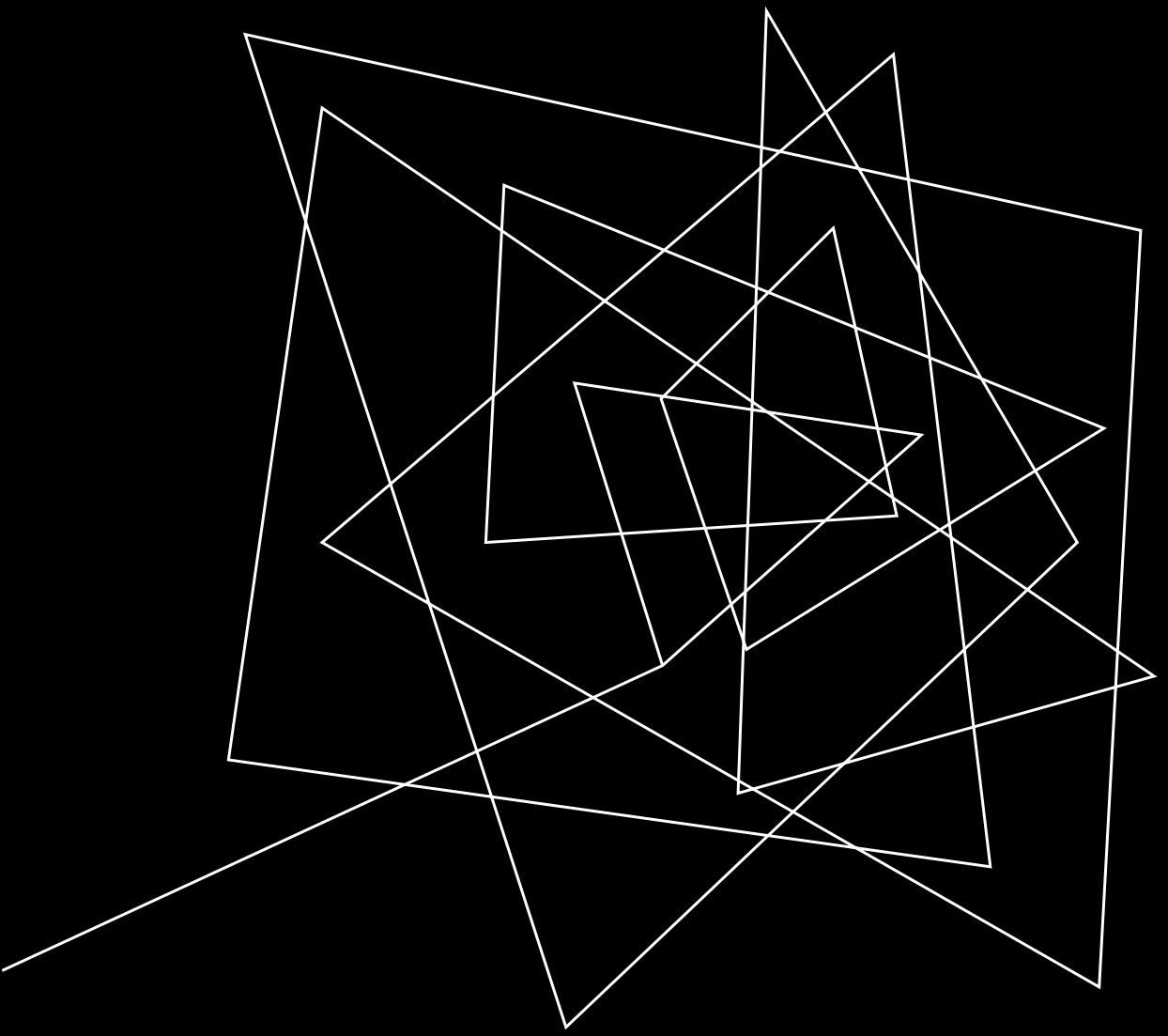


INTRODUCTION TO BIOINFORMATICS

IrBO 25 | By Mohammad Ebrahim Katebi

@Mekatebi



ALIGNMENT

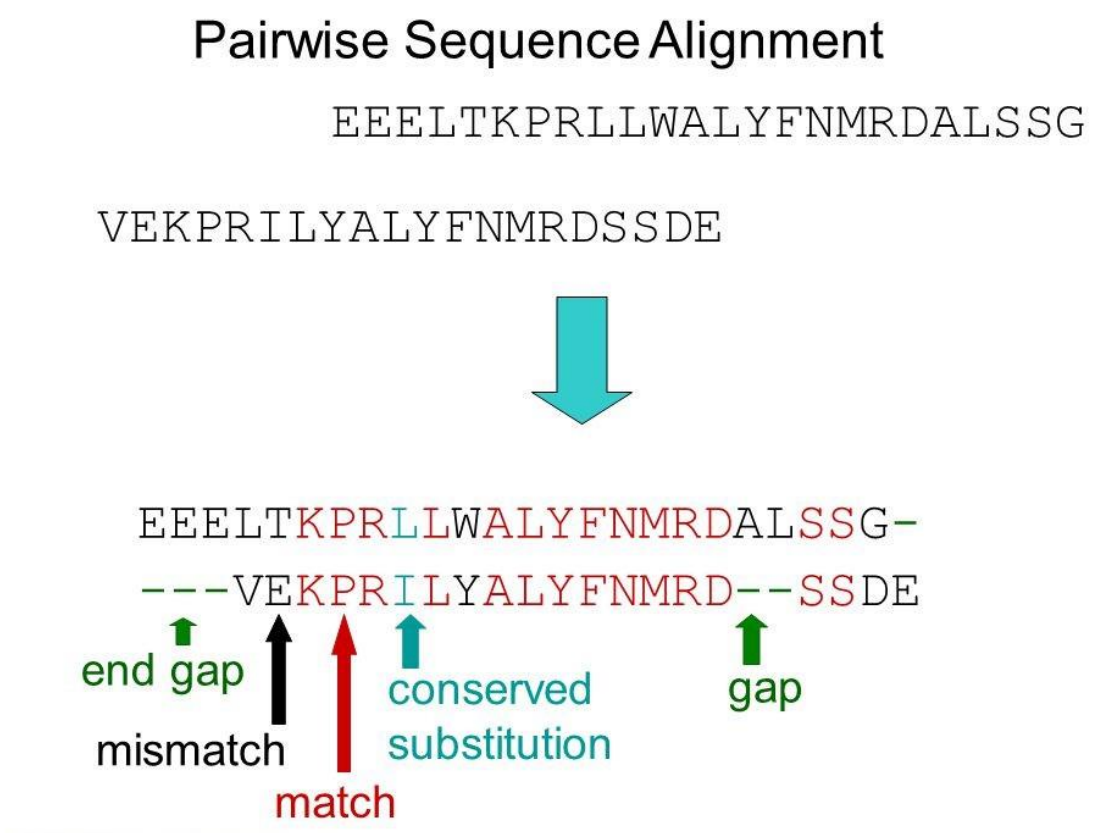
THE FOUNDATION OF BIOINFORMATICS

Why?

Conservation and Homology

- To obtain functional or mechanistic insight
- To find evolutionarily relation
- To find structurally or functionally similarity
- Similarity searching of databases
- Assembly of sequence reads

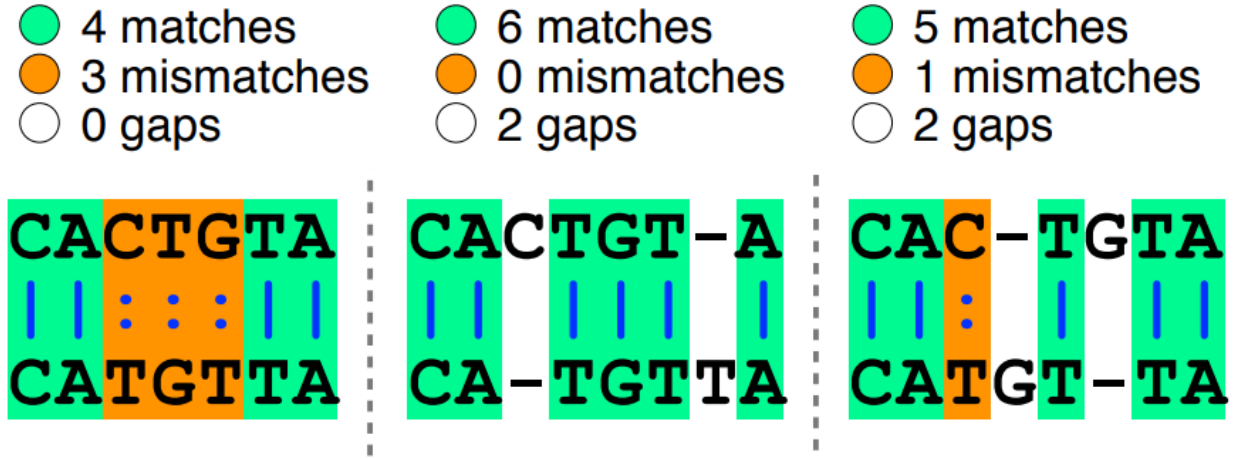
What?



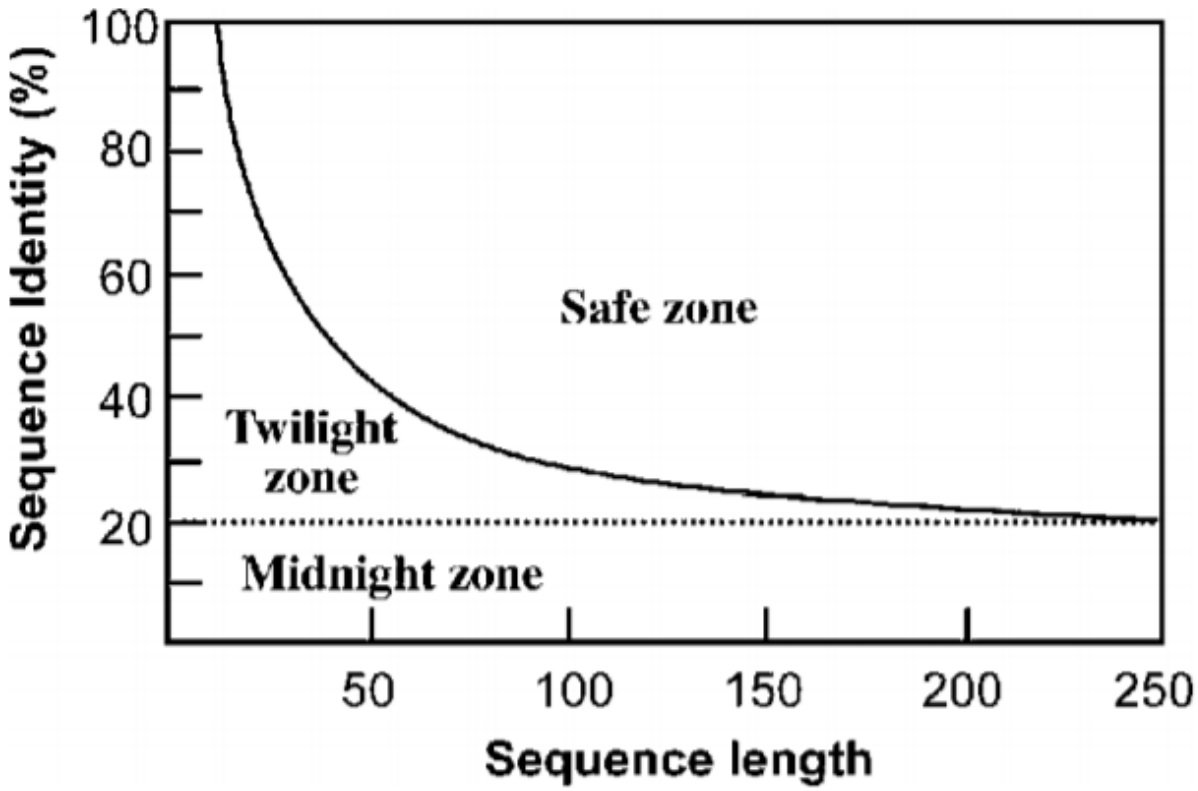
Similarity and Identity

Mutations, Deletions and Insertions

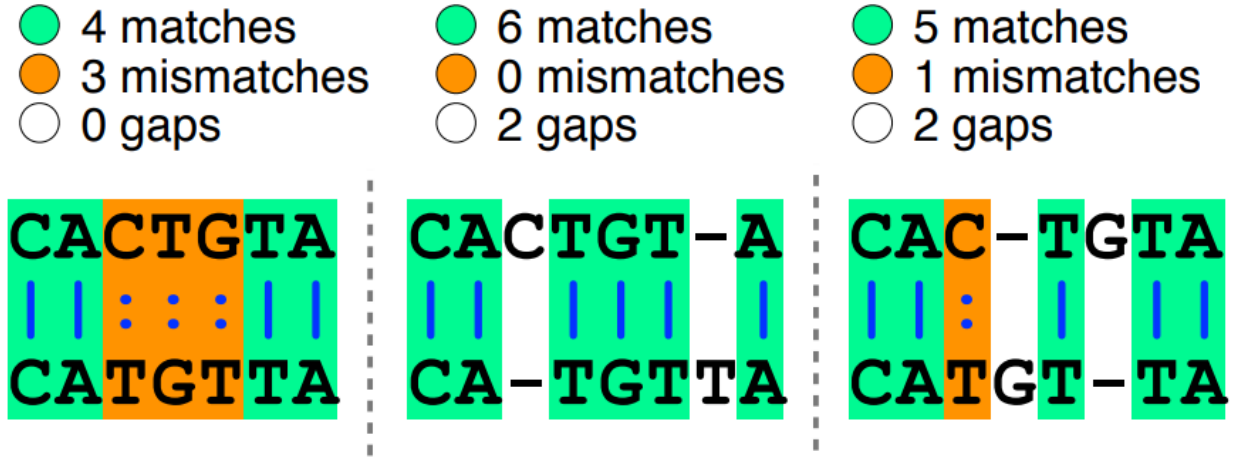
What is the best Alignment?



What is the best Alignment?



What is the best Alignment?

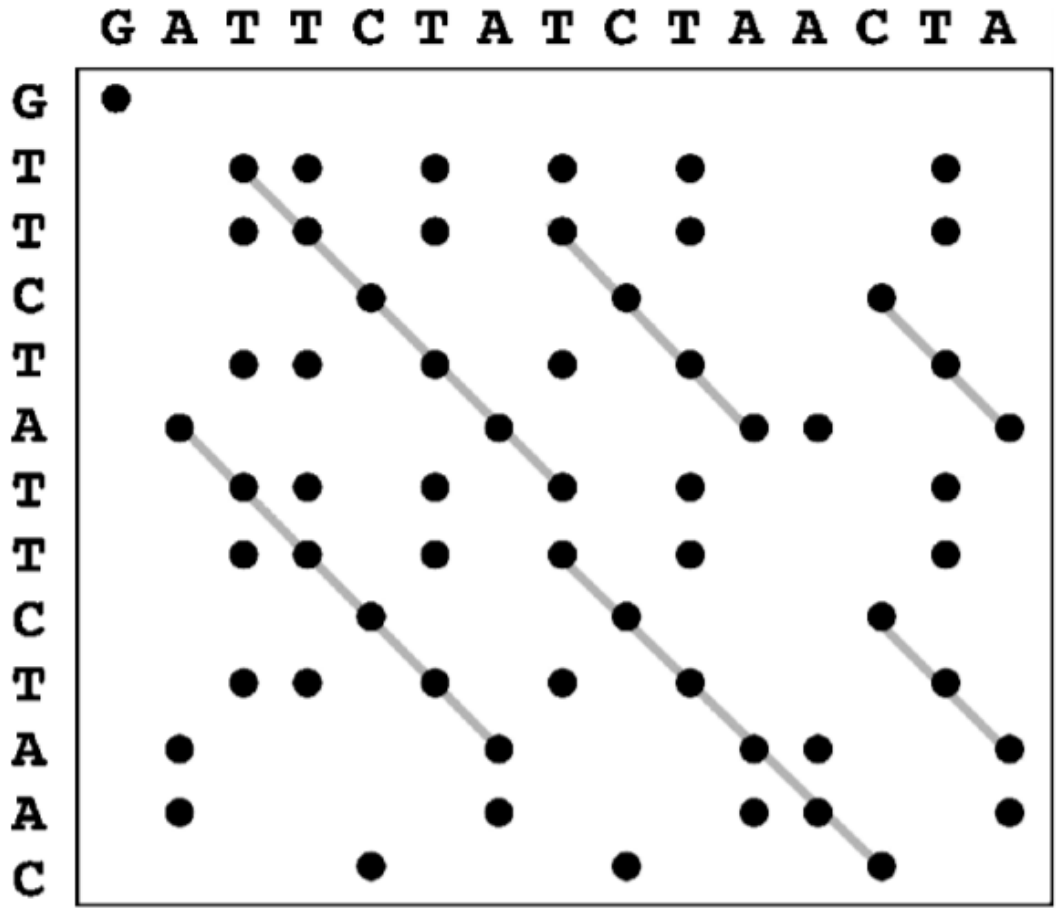


Scoring:

Match : +3
Mismatch : -1
Gap : -2

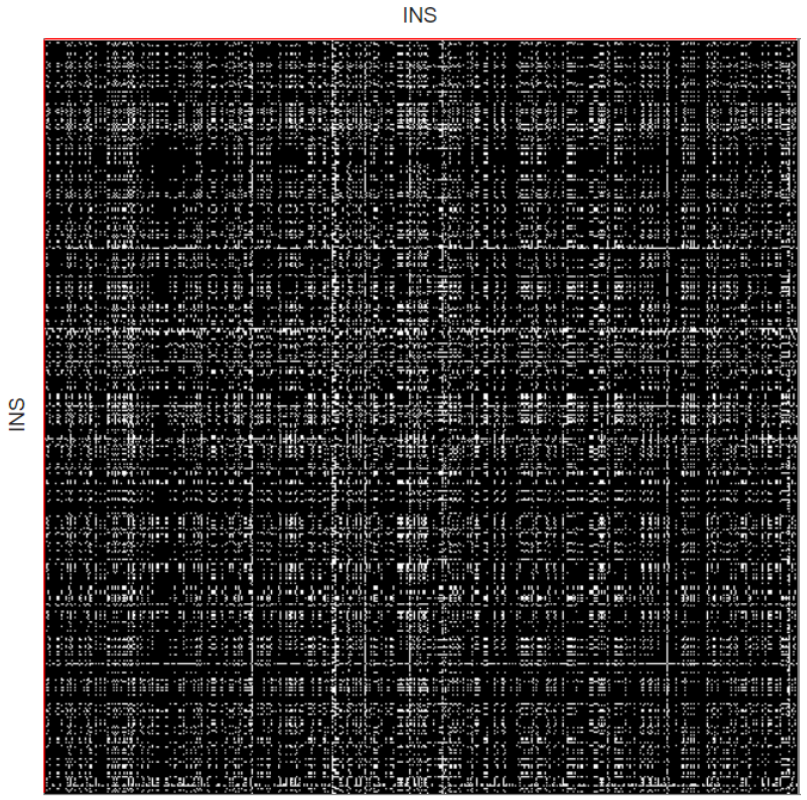
How?

Dot plots

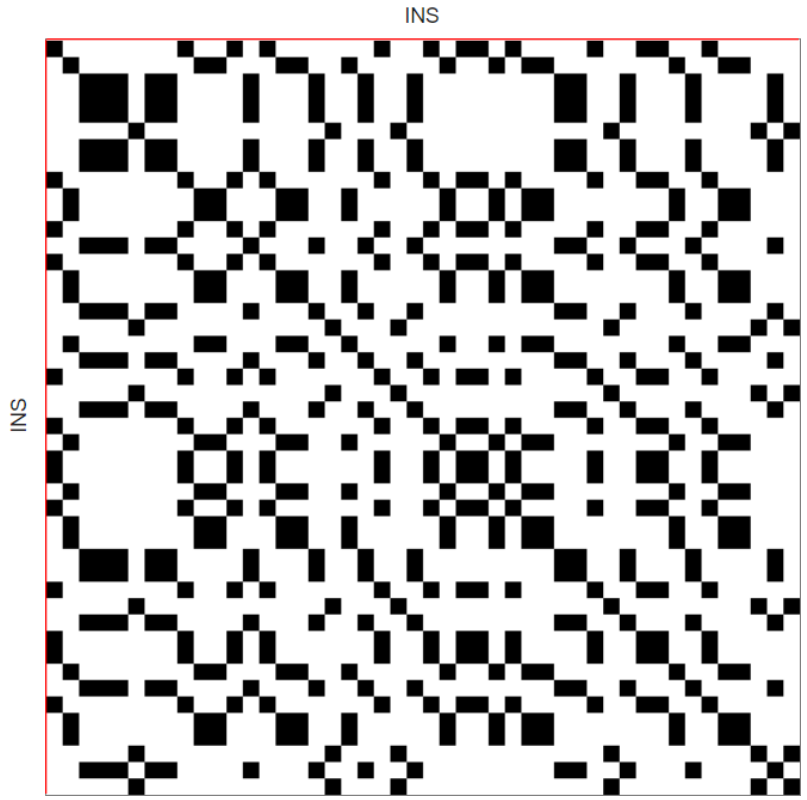

G A T T C T A T C T A A C T A
G
T
T
C
T
A
T
T
C
T
A
A
C

How a Dotplot of Homo sapiens insulin (INS) against itself would look like?

How a Dotplot of Homo sapiens insulin (INS) against itself would look like?

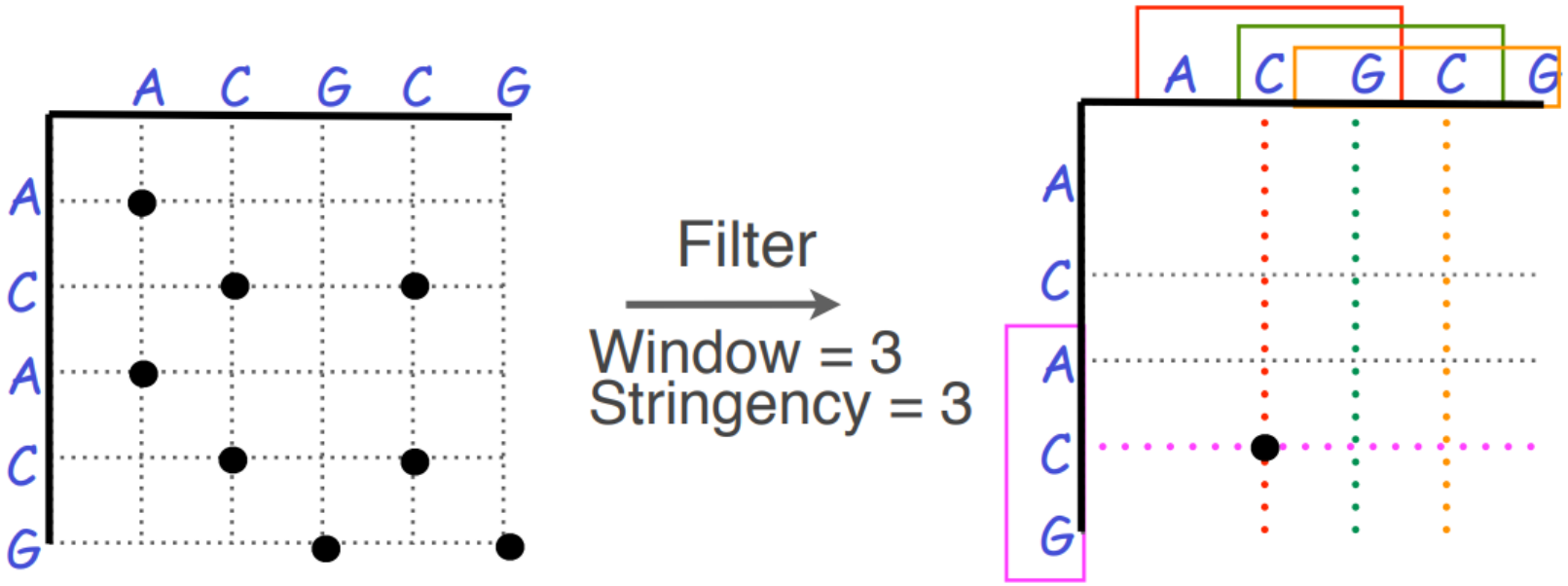


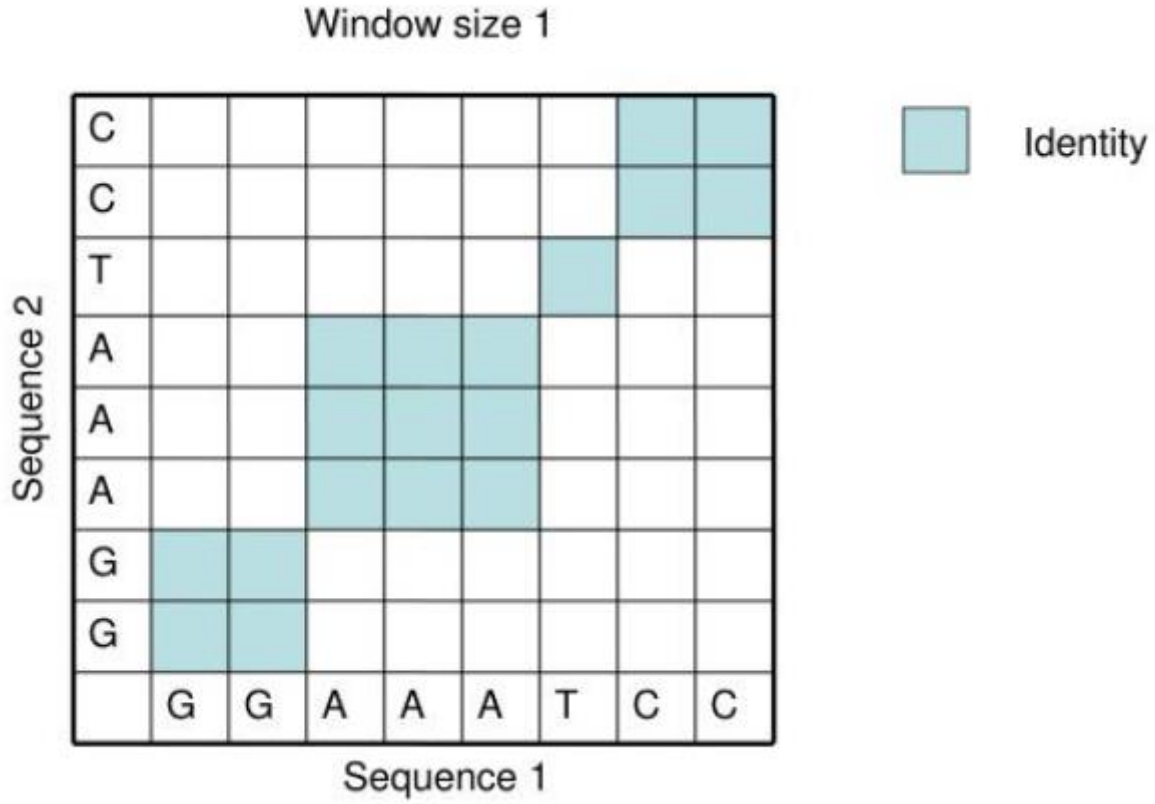
How a Dotplot of Homo sapiens insulin (INS) against itself would look like?

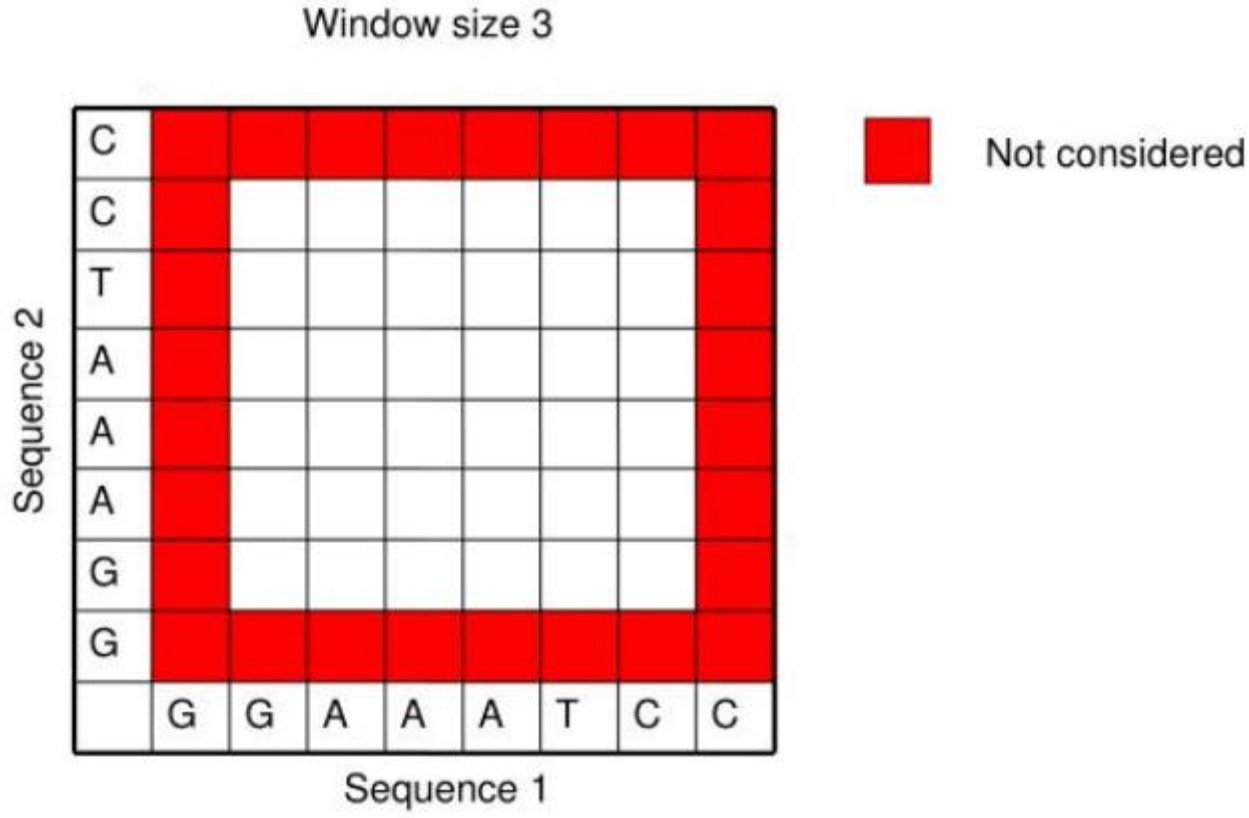


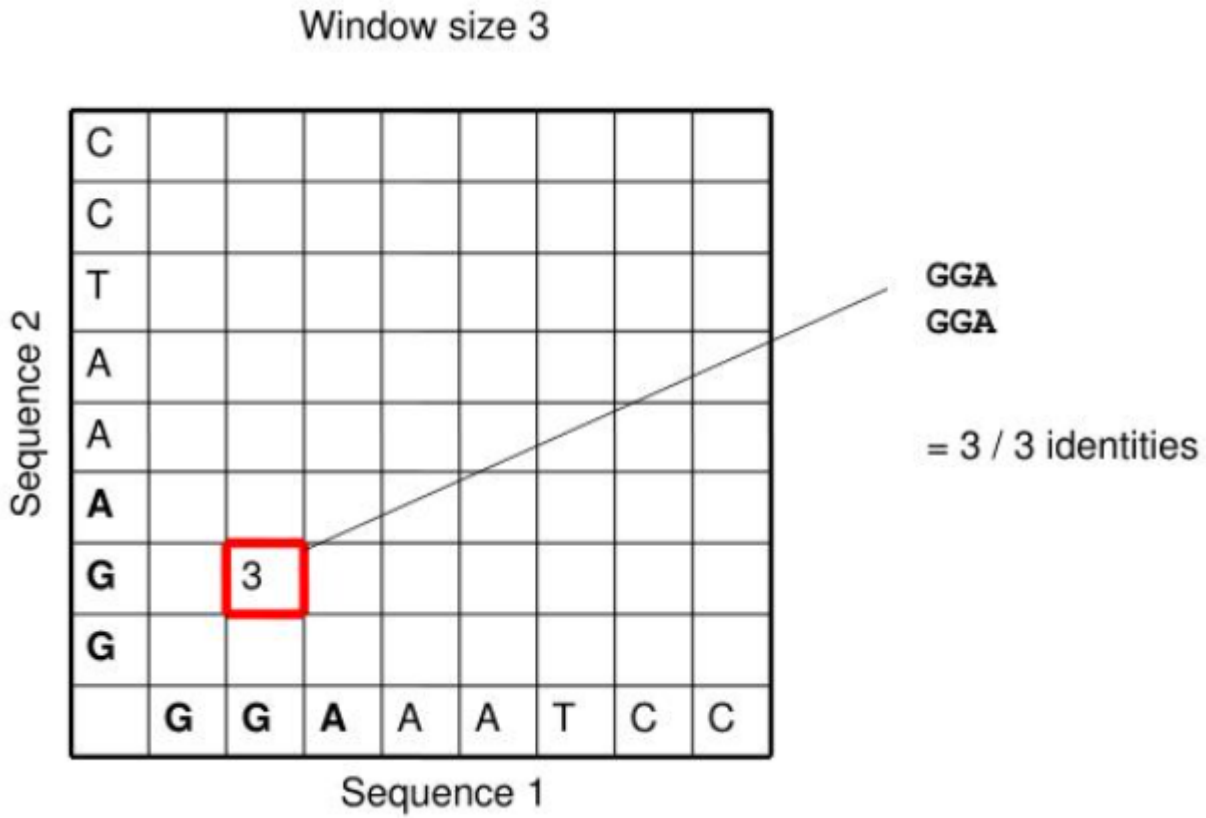
x1024

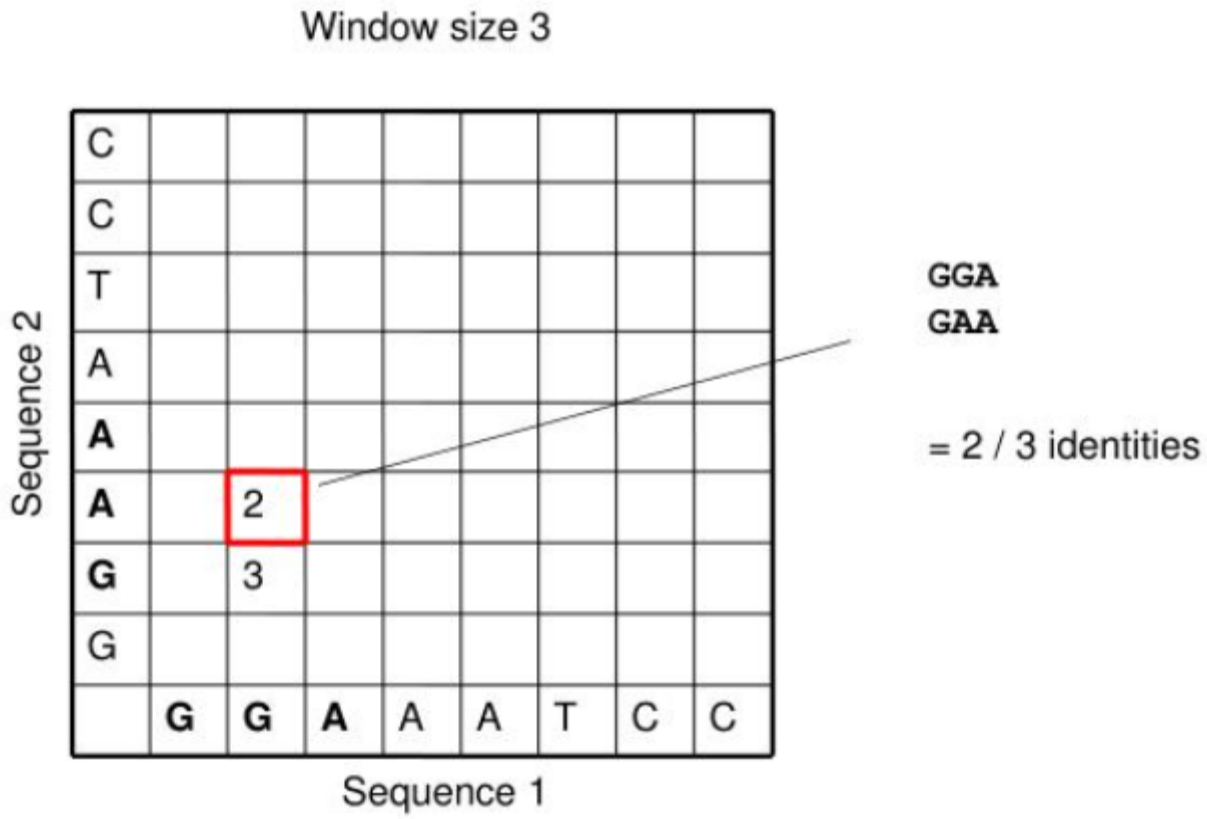
Noise reduction:
Window and Threshold



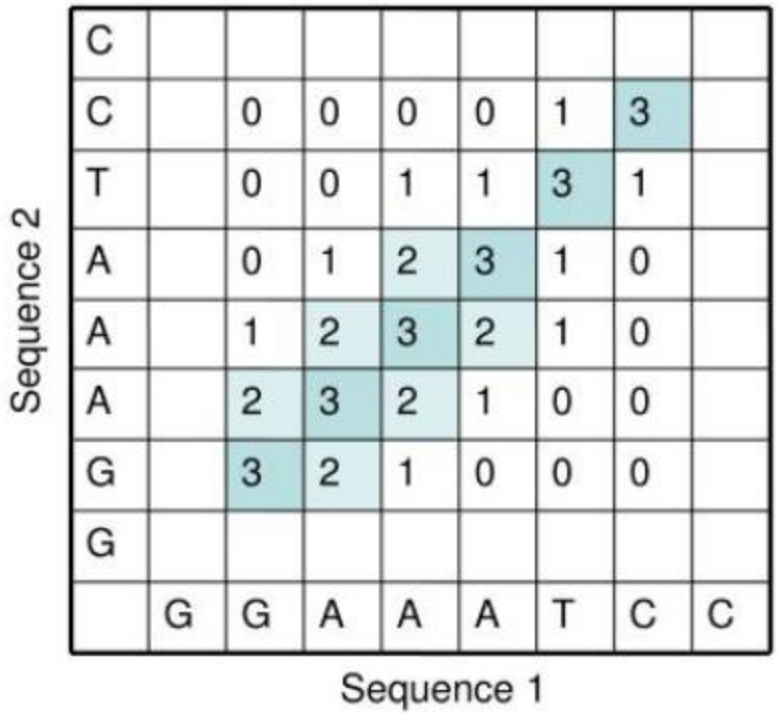

Window size 1
Sequence 2
C
C
T
A
A
A
G
G
G G A A A T C C
Sequence 1
Identity


Window size 3
Not considered
Sequence 2
C
C
T
A
A
A
G
G
G G A A A T C C
Sequence 1


Window size 3
Sequence 2
C
C
T
A
A
A
G
G
3
G G A A A T C C
Sequence 1
GGA
GGA
= 3 / 3 identities


Window size 3
Sequence 2
C
C
T
A
A
A
A
G
G
2
3
G G A A A T C C
Sequence 1
GGA
GAA
= 2 / 3 identities

Window size 3



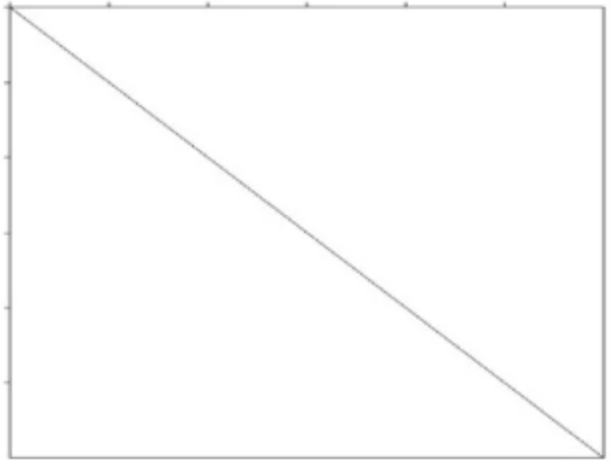
Identical sequence

- These are the two identical sequences:
  - Seq1: MALWGRL
  - Seq2: MALWGRL

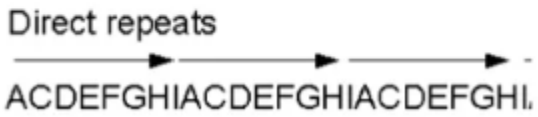
Identical sequence

- These are the two identical sequences:
  - Seq1: MALWGRL
  - Seq2: MALWGRL

	M	A	L	W	G	R	L
M	*						
A		*					
L			*				
W				*			
G					*		
R						*	
L							*



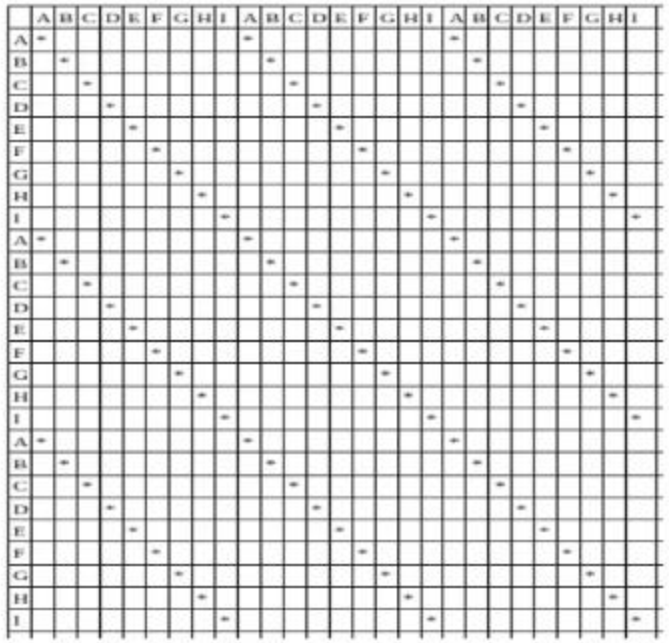
Direct repeat

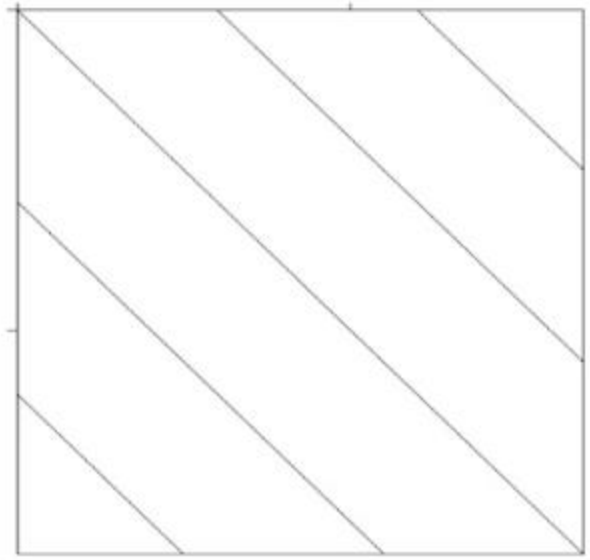


Direct repeat

Direct repeats

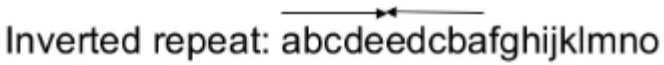
ACDEFGHIACDEFGHIACDEFGHI





Inverted repeat

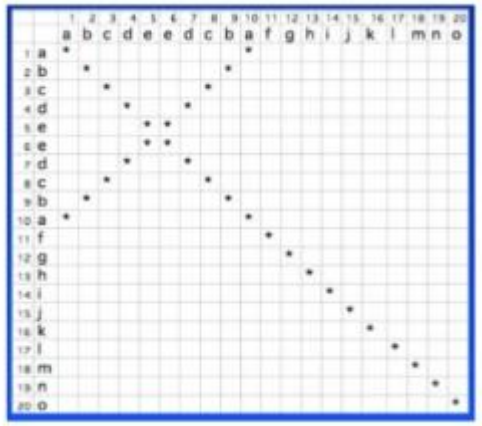
An inverted repeat is sequence of nucleotides followed downstream by its reverse complement.

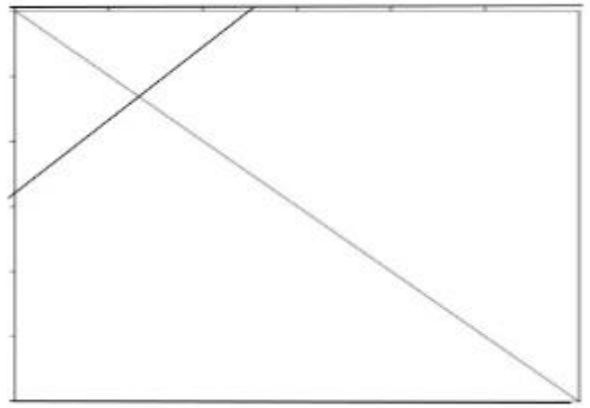


Inverted repeat

An inverted repeat is sequence of nucleotides followed downstream by its reverse complement.

Inverted repeat: abcdeedcbafghijklmno





Frame shifts

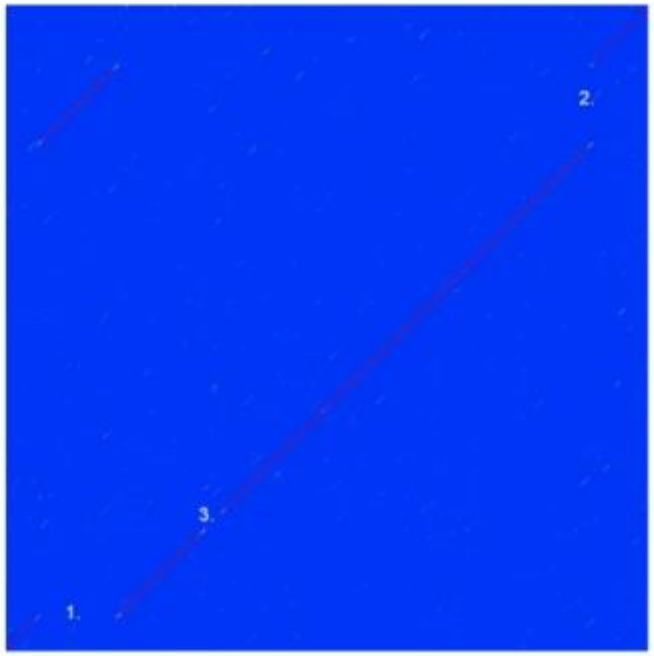
Frame shifts in a nucleotide sequence can occur due to insertions, deletions or mutations.

1. Deletion of nucleotides
2. Insertion of nucleotides
3. Mutation (out of frame)

Frame shifts

Frame shifts in a nucleotide sequence can occur due to insertions, deletions or mutations.

1. Deletion of nucleotides
2. Insertion of nucleotides
3. Mutation (out of frame)



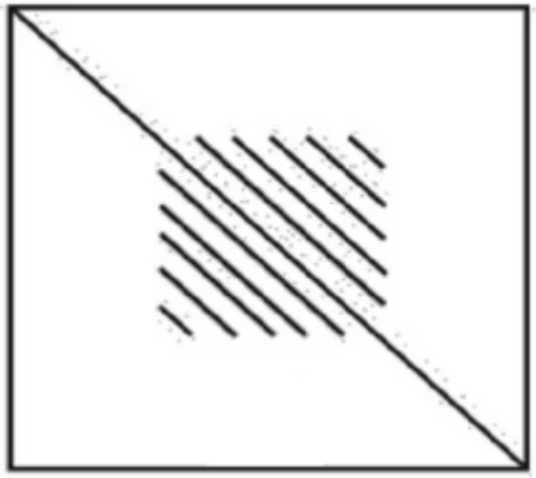
Low cmplexity region

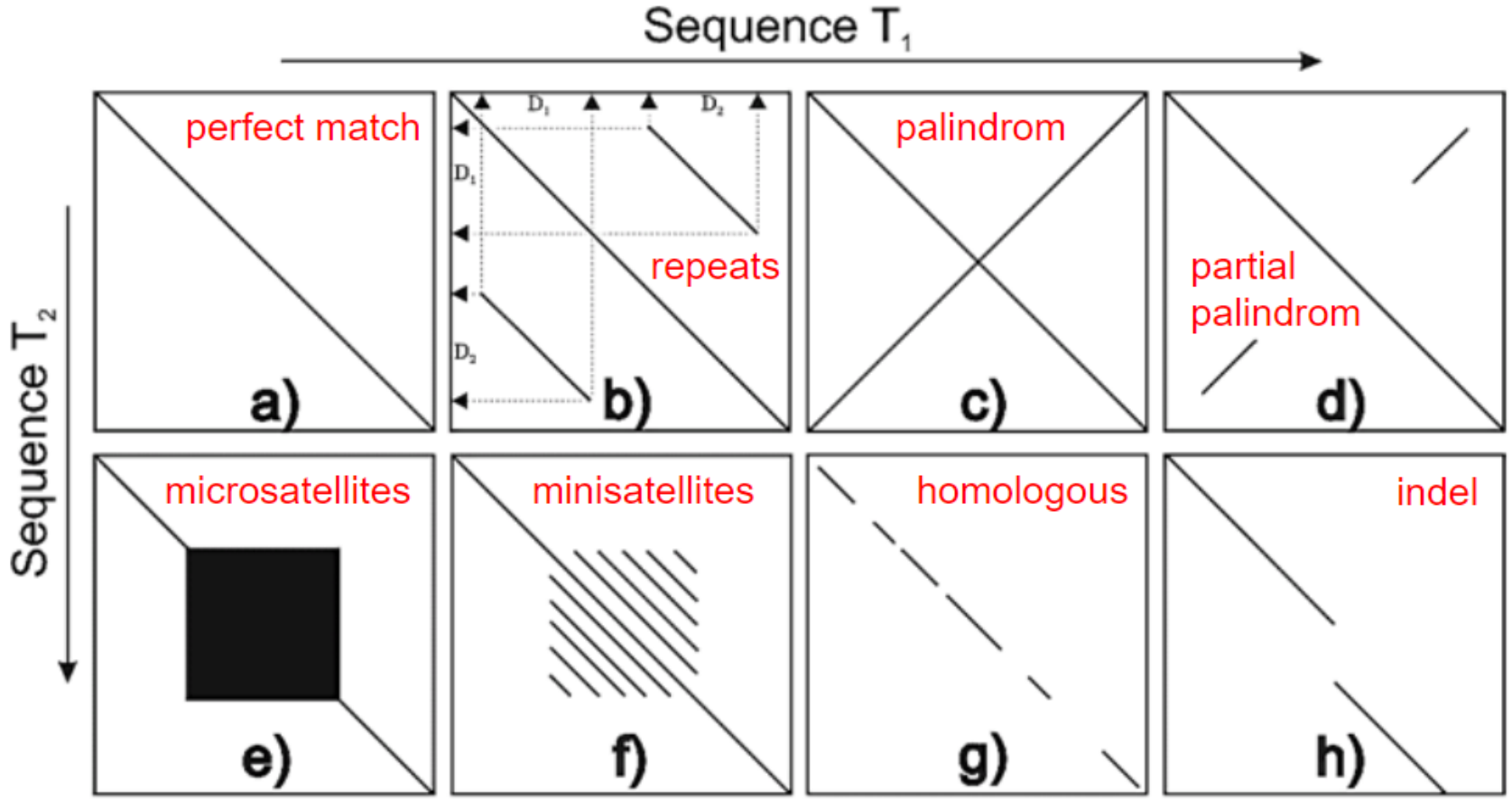
- Low-complexity regions in sequences can be found as regions around the diagonal all obtaining a high score. Low complexity regions are calculated from the redundancy of amino acids within a limited region [Wootton and Federhen,1993].

Low cmplexity region

- Low-complexity regions in sequences can be found as regions around the diagonal all obtaining a high score. Low complexity regions are calculated from the redundancy of amino acids within a limited region [Wootton and Federhen,1993].

	C	D	E	E	E	E	E	F	G
C	o								
D		o							
E			o	o	o	o	o		
E			o	o	o	o	o		
E			o	o	o	o	o		
E			o	o	o	o	o		
E			o	o	o	o	o		
F								o	
G									o

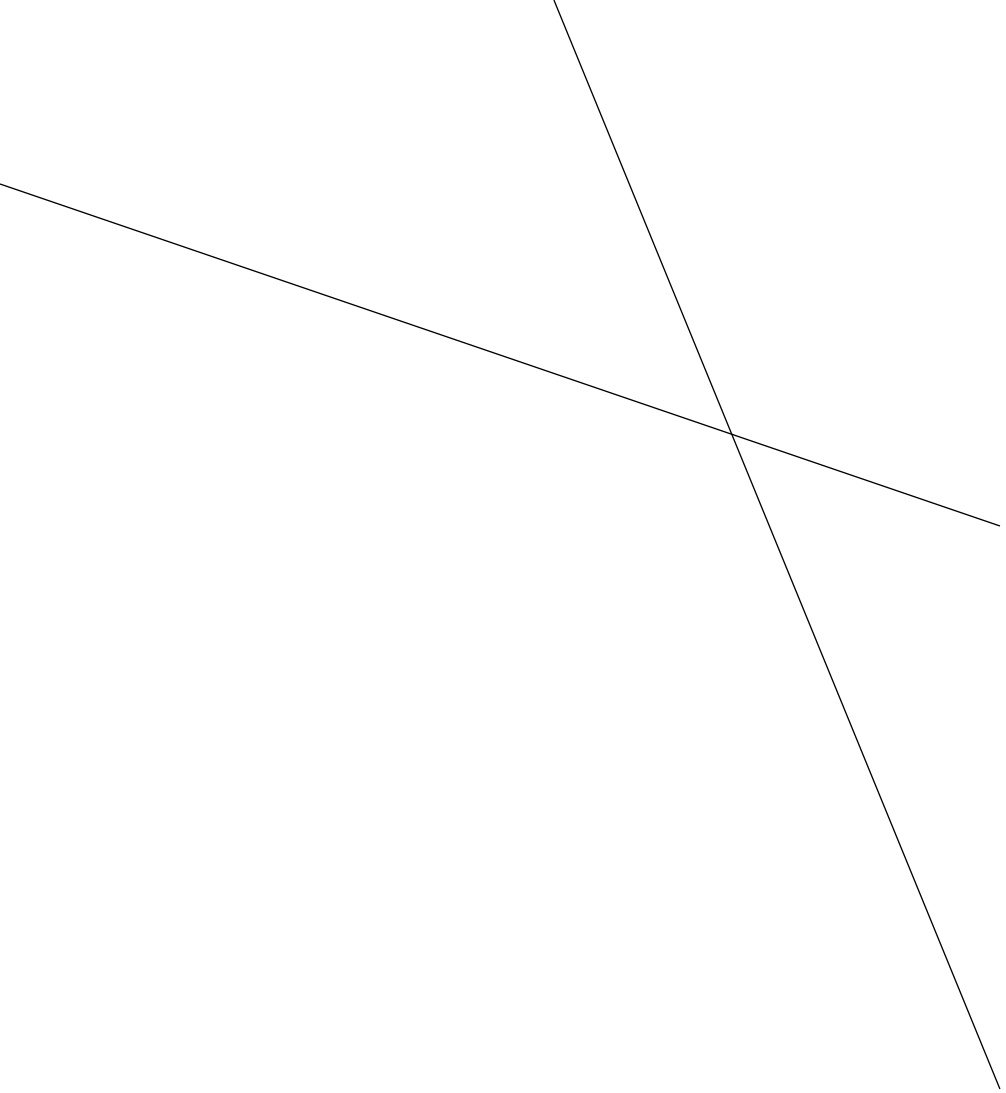



Sequence T_1
Sequence T_2
perfect match
a)
D_1
D_2
D_1
D_2
repeats
b)
palindrom
c)
partial
palindrom
d)
microsatellites
e)
minisatellites
f)
homologous
g)
indel
h)

Your Turn! :)

https://dotlet.vital-it.ch/

https://www.ebi.ac.uk/Tools/seqstats/

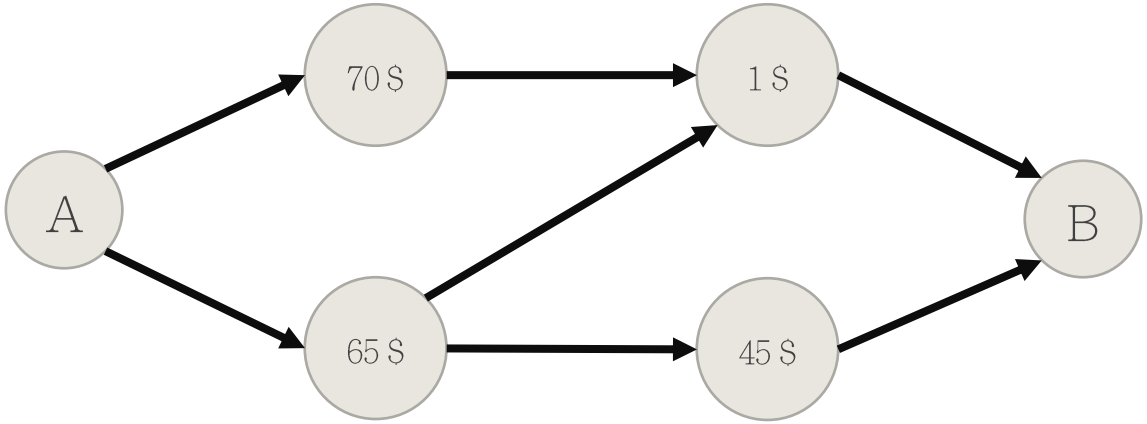


How?

Dynamic Programming

Optimization:

- Exact algorithms
- Heuristic algorithms (Greedy Algorithms)



Dynamic Programming: simplifying a complicated problem by breaking it down into simpler sub-problems.

Dynamic programming may be used to solve problems with:

- ❖ Optimal Substructure: The optimal solution to an instance of the problem contains optimal solutions to subproblems.

- ❖ Overlapping Subproblems: There are a limited number of subproblems, many/most of which are repeated many times.

- ✓ Alignment scores are additive.

This allows us to create a **matrix** M indexed by i and j, which are positions in two sequences S and T to be aligned. The best alignment of S and T corresponds with the best path through the matrix M after it is filled in using a recursive formula.

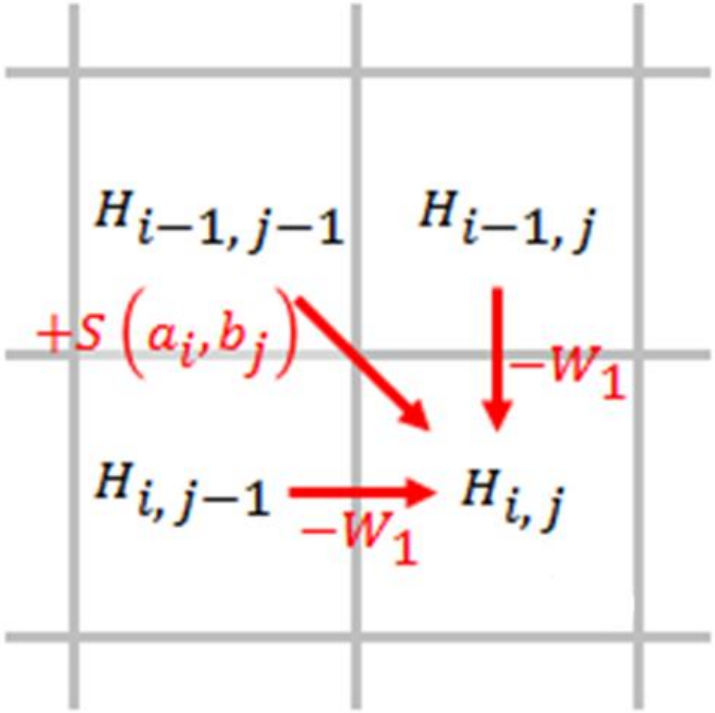
The Needleman—Wunsch algorithm

Scoring:

Match : +3
Mismatch : -1
Gap : -2

		A	G	T	T	C
	0					
A						
T						
T						
G						
C						

The Needleman—Wunsch algorithm



$$F_{ij} = \max(F_{i-1,j-1} + S(A_i, B_j),\ F_{i,j-1} + d,\ F_{i-1,j} + d)$$

The Needleman—Wunsch algorithm

Scoring:

Match : +3
Mismatch : -1
Gap : -2

		A	G	T	T	C
	0					
A						
T						
T						
G						
C						

The Needleman—Wunsch algorithm

Scoring:

Match : +3
Mismatch : -1
Gap : -2

		A	T	T	G	C
	0	-2	-4	-6	-8	-10
A	-2	3	1	-1	-3	-5
G	-4	1	2	0	2	0
T	-6	-1	4	5	3	1
T	-8	-3	2	7	5	3
C	-10	-5	0	5	6	8

The Needleman—Wunsch algorithm

Scoring:

Match : +3
Mismatch : -1
Gap : -2

		A	T	T	G	C
	0	-2	-4	-6	-8	-10
A	-2	3	1	-1	-3	-5
G	-4	1	2	0	2	0
T	-6	-1	4	5	3	1
T	-8	-3	2	7	5	3
C	-10	-5	0	5	6	8

The Needleman—Wunsch algorithm

Scoring:

Match : +3
Mismatch : -1
Gap : -2

		A	T	T	G	C
	0	-2	-4	-6	-8	-10
A	-2	3	1	-1	-3	-5
G	-4	1	2	0	2	0
T	-6	-1	4	5	3	1
T	-8	-3	2	7	5	3
C	-10	-5	0	5	6	8

A - T T G C
A G T T - C
Score = 8

Your Turn! :)

Seq 1: TTAGCG
Seq 2: AAGCTG

The **Needleman—Wunsch** algorithm

Scoring

Match : +3
Mismatch : -1
Gap : -1

Your Turn! :)

Seq 1: TTAGCG
Seq 2: AAGCTG

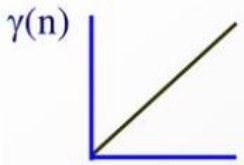
The **Needleman—Wunsch** algorithm

http://rna.informatik.uni-freiburg.de/Teaching/

Scoring

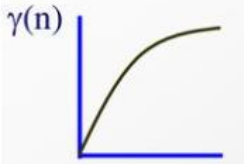
Match : +3
Mismatch : -1
Gap : -1

Generalized gap penalty



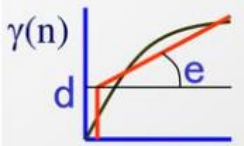
Linear gap penalty: w(k) = k*p

- State: Current index tells if in a gap or not
- Achievable using quadratic algorithm (even w/ linear space)



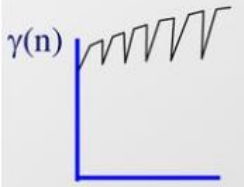
Quadratic: $w(k) = p+q*k+rk^2$.

- State: needs to encode the length of the gap, which can be O(n)
- To encode it we need O(log n) bits of information. Not feasible



Affine gap penalty: w(k) = p + q*k, where q<p

- State: add binary value for each sequence: starting a gap or not
- Implementation: add second matrix for already-in-gap (recitation)

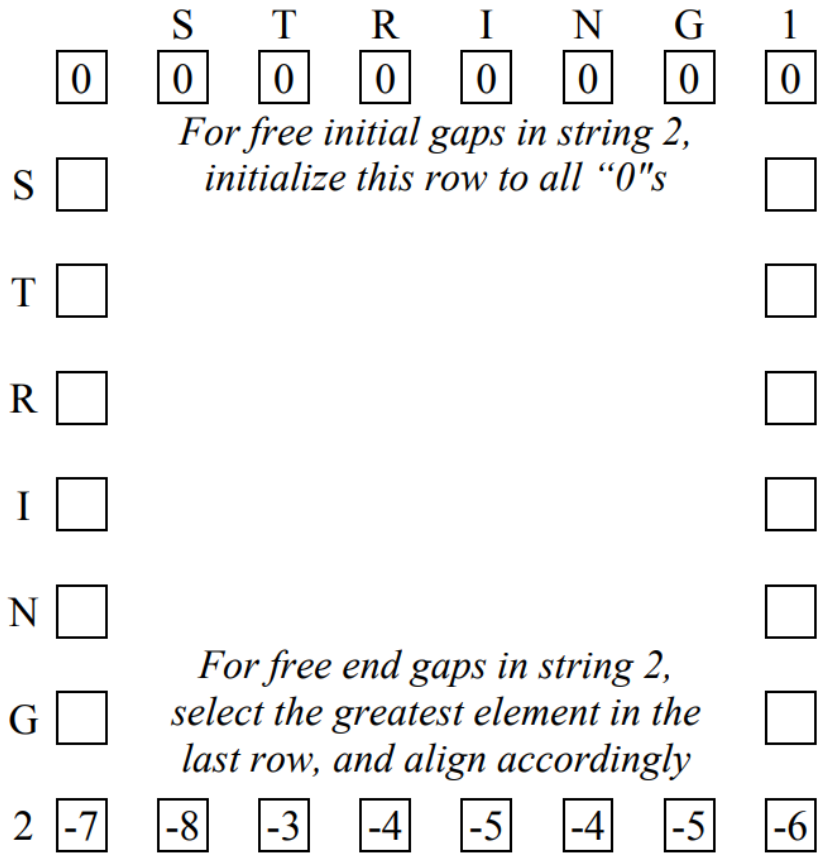


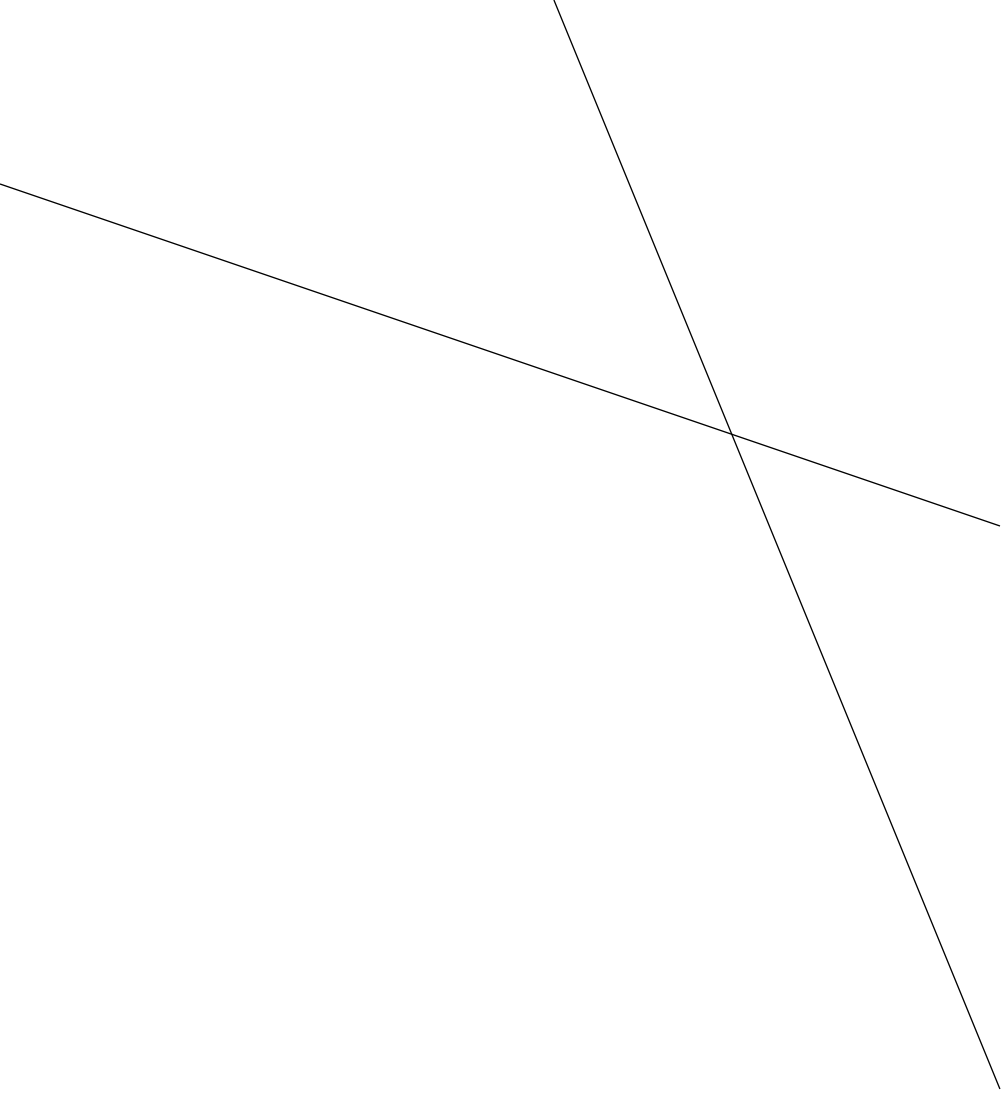
Length (mod 3) gap penalty for protein-coding regions

- Gaps of length divisible by 3 are penalized less: conserve frame
- This is feasible, but requires more possible states
- Possible states are: starting, mod 3=1, mod 3=2, mod 3=0

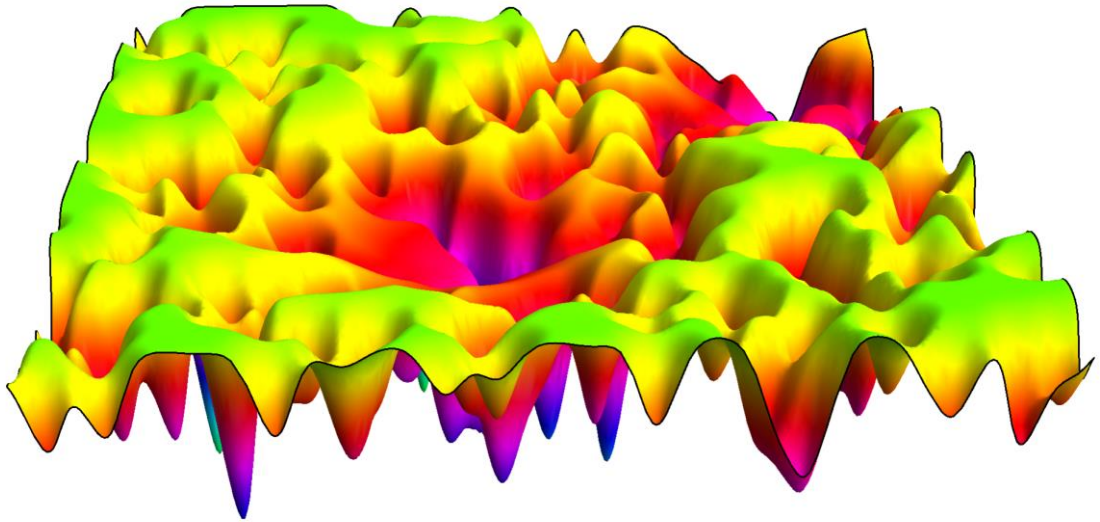
Semi-global Alignment

Semi-Global Alignment

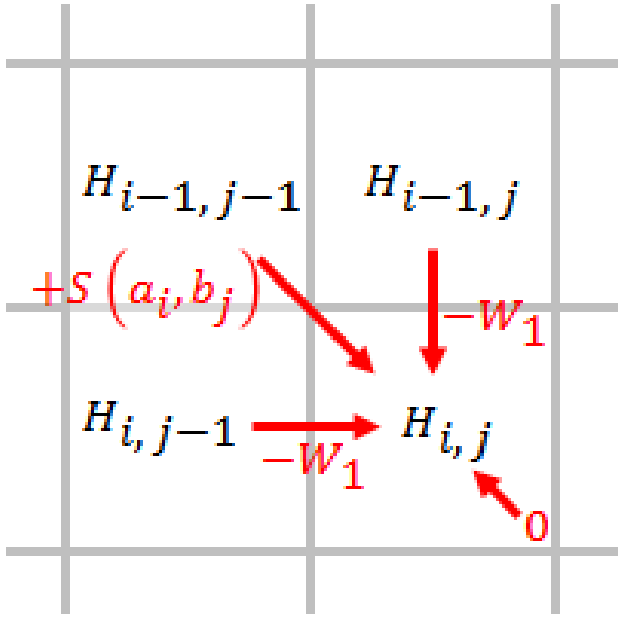




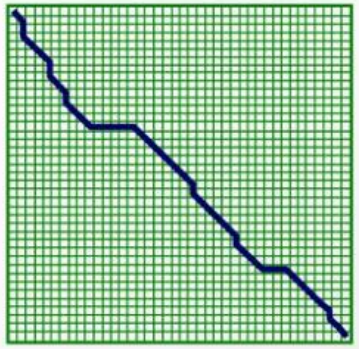
Local Alignment



The Smith—Waterman algorithm



$$H_{ij} = \max \begin{cases} H_{i-1,j-1} + s(a_i, b_j), \\ H_{i-1,j} - W_1, \\ H_{i,j-1} - W_1, \\ 0 \end{cases}$$



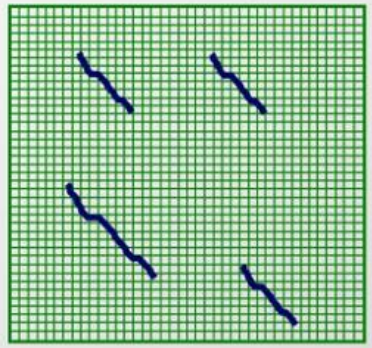
Needleman-Wunsch algorithm

Initialization: $F(0, 0) = 0$

Iteration:

$$F(i, j) = \max \begin{cases} F(i-1, j) - d \\ F(i, j-1) - d \\ F(i-1, j-1) + s(x_i, y_j) \end{cases}$$

Termination: Bottom right



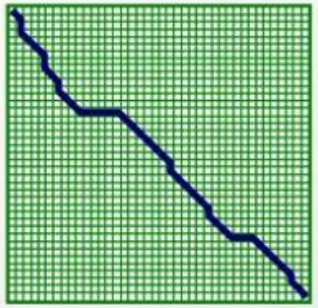
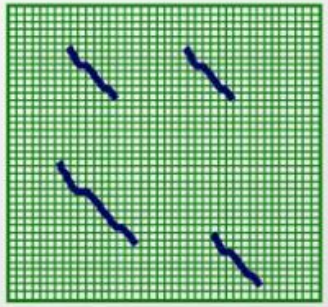
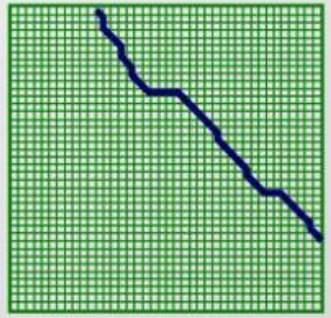
Smith-Waterman algorithm

Initialization: $F(0, j) = F(i, 0) = 0$

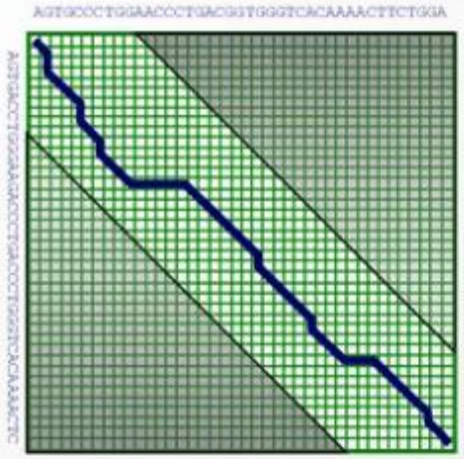
Iteration:

$$F(i, j) = \max \begin{cases} 0 \\ F(i-1, j) - d \\ F(i, j-1) - d \\ F(i-1, j-1) + s(xi, yj) \end{cases}$$

Termination: Anywhere

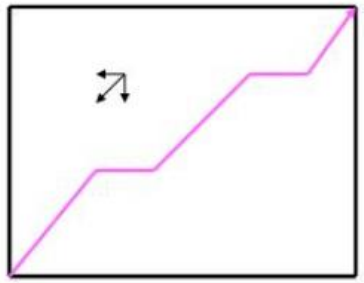




	Global	Local	Semi-global
Initialization	Top left	Top row/left col.	Top row
Iteration:max	$F(i-1, j)-d$ $F(i, j-1)-d$ $F(i-1, j-1)+s(x_i, y_j)$	0 $F(i-1, j)-d$ $F(i, j-1)-d$ $F(i-1, j-1)+s(x_i, y_j)$	$F(i-1, j)-d$ $F(i, j-1)-d$ $F(i-1, j-1)+s(x_i, y_j)$
Termination	Bottom right	Anywhere	Right column

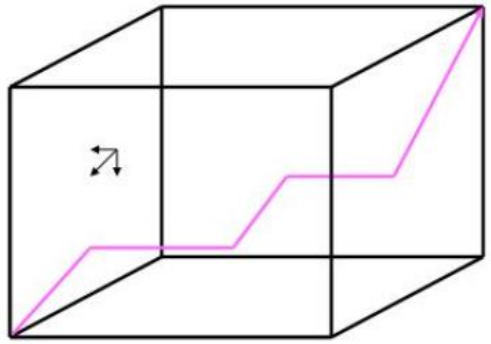


- Save time: Bounded-space computation
  - Space: O(k*m)
  - Time: O(k*m), where k = radius explored
  - Heuristic
    - Not guaranteed optimal answer
    - Works very well in practice
  - Practical interest

Multiple Sequence Alignment (MSA)



Finding a PSA = Finding a path through a 2-Dim matrix.



Finding an MSA = Finding a path through an N-Dim matrix. It's $O(L^N)$, where N is the number of sequences and L is the sequence length.

Scoring

Construction of Substitution Matrices

The idea behind the scoring matrix is that the score of alignment should reflect the probability that two similar sequences are homologous.

- Hypothesis 1: That the alignment between the two sequence is due to chance and the sequences are, in fact, unrelated. (U)

- Hypothesis 2: That the alignment is due to common ancestry and the sequences are actually related. (R)

For **unrelated** sequences, the probability of having an n-residue alignment between x and y is a simple product of the probabilities of the individual sequences since the residue pairings are independent.

For **unrelated** sequences, the probability of having an n-residue alignment between x and y is a simple product of the probabilities of the individual sequences since the residue pairings are independent.

$$\mathbf{x} = \{x_1 \dots x_n\}$$
$$\mathbf{y} = \{y_1 \dots x_n\}$$
$$q_a = P(\text{ amino acid } a)$$
$$P(\mathbf{x}, \mathbf{y} \mid U) = \prod_{i=1}^{n} q_{x_i} \prod_{i=1}^{n} q_{y_i}$$

For **related** sequences, the residue pairings are no longer independent so we must use a different joint probability, assuming that each pair of aligned amino acids evolved from a common ancestor:

$$p_{ab} = P(\text{ evolution gave rise to } a \text{ in } \mathbf{x} \text{ and } b \text{ in } \mathbf{y})$$

$$P(\mathbf{x}, \mathbf{y} \mid R) = \prod_{i=1}^{n} p_{x_i y_i}$$

A principled way of comparing R and U:
The Likelihood Ratio between the two

$$\frac{P(\mathbf{x}, \mathbf{y} \mid R)}{P(\mathbf{x}, \mathbf{y} \mid U)} = \frac{\prod_{i=1}^{n} p_{x_i y_i}}{\prod_{i=1}^{n} q_{x_i} \prod_{i=1}^{n} q_{y_i}}$$

$$= \frac{\prod_{i=1}^{n} p_{x_i y_i}}{\prod_{i=1}^{n} q_{x_i} q_{y_i}}$$

Since we eventually want to compute a sum of scores and probabilities require add products, we take the log of the products.

$$\log \frac{\Pr(x, y \mid R)}{\Pr(x, y \mid U)} = \sum_i \log\left(\frac{p_{x_i y_i}}{q_{x_i} q_{y_i}}\right)$$

We define the alignment score as the log of the Likelihood Ratio:

$$S = \sum_i s(x_i, y_i) = \log \frac{\Pr(x, y \mid R)}{\Pr(x, y \mid U)} = \sum_i \log\left(\frac{p_{x_i y_i}}{q_{x_i} q_{y_i}}\right)$$

The substitution matrix score for a given pair a,b:

$$s(a,b) = \log\left(\frac{p_{ab}}{q_a q_b}\right)$$

Hypothesis we wish to test; two amino acids are correlated because they are homologous.

$$\text{score} = \log \text{ odds ratio} = s_{AB} \propto \log\left(\frac{\text{observed}}{\text{expected}}\right)$$

Null hypothesis; two amino acids occur independently (and are uncorrelated and unrelated).

But how do we get P_{ab}?

!

Key Idea:

Empirical observations

Research first done by Margaret Dayhoff in 1970s and colleagues and later by Henikoff and Henikoff early 1990s resulted in PAM and BLOSUM Substitution matrices and are the most widely used today.

BLOSUM Matrices

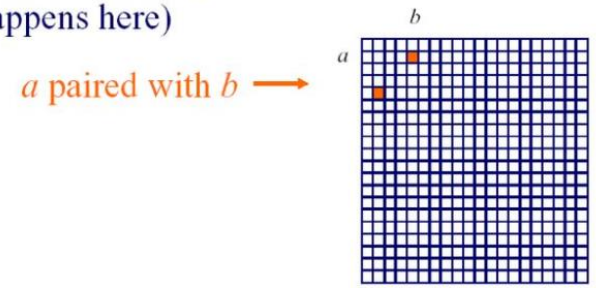
BLOcks SUbstitution Matrix

BLOSUM Matrices

- [Henikoff & Henikoff, *PNAS* 1992]
- probabilities estimated from "blocks" of sequence fragments that represent structurally conserved regions in proteins
- transition frequencies observed directly by identifying blocks that are at least
  - 45% identical (BLOSUM-45)
  - 50% identical (BLOSUM-50)
  - 62% identical (BLOSUM-62)
  - etc.

BLOSUM Matrices

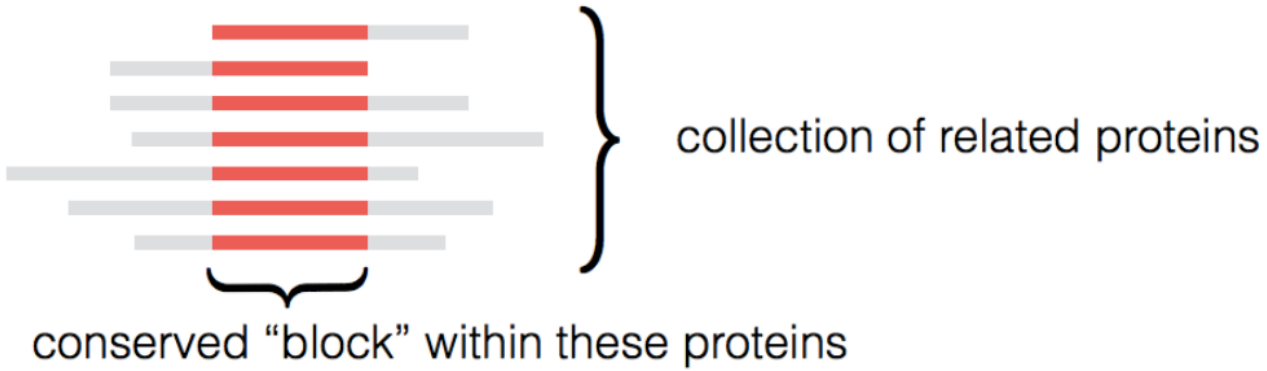
- given: a set of sequences in a block
- fill in matrix A with number of observed substitutions (we won't worry about details of some normalization that happens here)



$$p_{ab} = \frac{A_{ab}}{\sum_{c,d} A_{cd}} \qquad q_a = \frac{\sum_b A_{ab}}{\sum_{c,d} A_{cd}}$$

The Blocks Database

1. Look for conserved (gapless) regions in alignments.



Henikoff and Henikoff analyzed conserved regions of related protein sequences they obtained from BLOCKS database. In total, they examined 2,000 blocks without gaps and 500 groups of related proteins by counting the number of matches and mismatches of each type of the 20 different amino acids.

2. Count all pairs of amino acids in each column of the alignments.

FPTADAGGRS
FVTADALGRS
FPTPDAGLRN
FVTAEAGIRQ
FPTAEAGGRS

$$c_{AB}^{(i)} = \begin{cases} \binom{c_A^{(i)}}{2} & \text{if } A = B \\ c_A^{(i)} \times c_B^{(i)} & \text{otherwise} \end{cases}$$

$c_A^{(i)} = \text{num. of occurrences of } A \text{ in column } i$

2. Count all pairs of amino acids in each column of the alignments.

FPTADAGGRS
FVTADALGRS
FPTPDAGLRN
FVTAEAGLRQ
FPTAEAGGRS

Example:

$$c_{GG}^{(i)} = \binom{3}{2} = 3$$

$$c_{GL}^{(i)} = 3 \times 2$$

$$c_{LL}^{(i)} = \binom{2}{2} = 1$$

In this column, there are 3 ways to pair G with G, 6 potential ways to pair G with L and 1 potential way to pair L with L.

3. Use amino acid pair frequencies to derive “score” for a mutation/replacement

Total # of potential align. between A & B: $c_{AB} = \sum_i c_{AB}^{(i)}$

Total number of pairwise char. alignments: $T = \sum_{A \geq B} c_{AB}$

Normalized frequency of aligning A & B: $q_{AB} = \frac{c_{AB}}{T}$

3. Use amino acid pair frequencies to derive “score” for a mutation/replacement

Probability of occurrence of amino acid A in any {A,B} pair:

$$p_A = q_{AA} + \sum_{A \neq B} q_{AB}$$

Expected likelihood of each {A,B} pair, assuming independence:

$$e_{AB} = \begin{cases} (p_A)(p_B) = (p_A)^2 & \text{if } A = B \\ (p_A)(p_B) + (p_B)(p_A) = 2(p_A)(p_B) & \text{otherwise} \end{cases}$$

To get the final scores in the matrix, Henikoff and Henikoff further converted the Log−Odds Ratios into bit units and multiplied each bit score by a scaling factor of two and rounded to the nearest integer, producing the final scores in BLOSUM matrix.

$$\text{score} = \log \text{ odds ratio} = s_{AB} \propto \log\left(\frac{\text{observed}}{\text{expected}}\right)$$

$$\text{score = log odds ratio = } s_{AB} = \text{Round}\left(\left(\frac{1}{\lambda}\right)\log_2\left(\frac{q_{AB}}{e_{AB}}\right)\right)$$

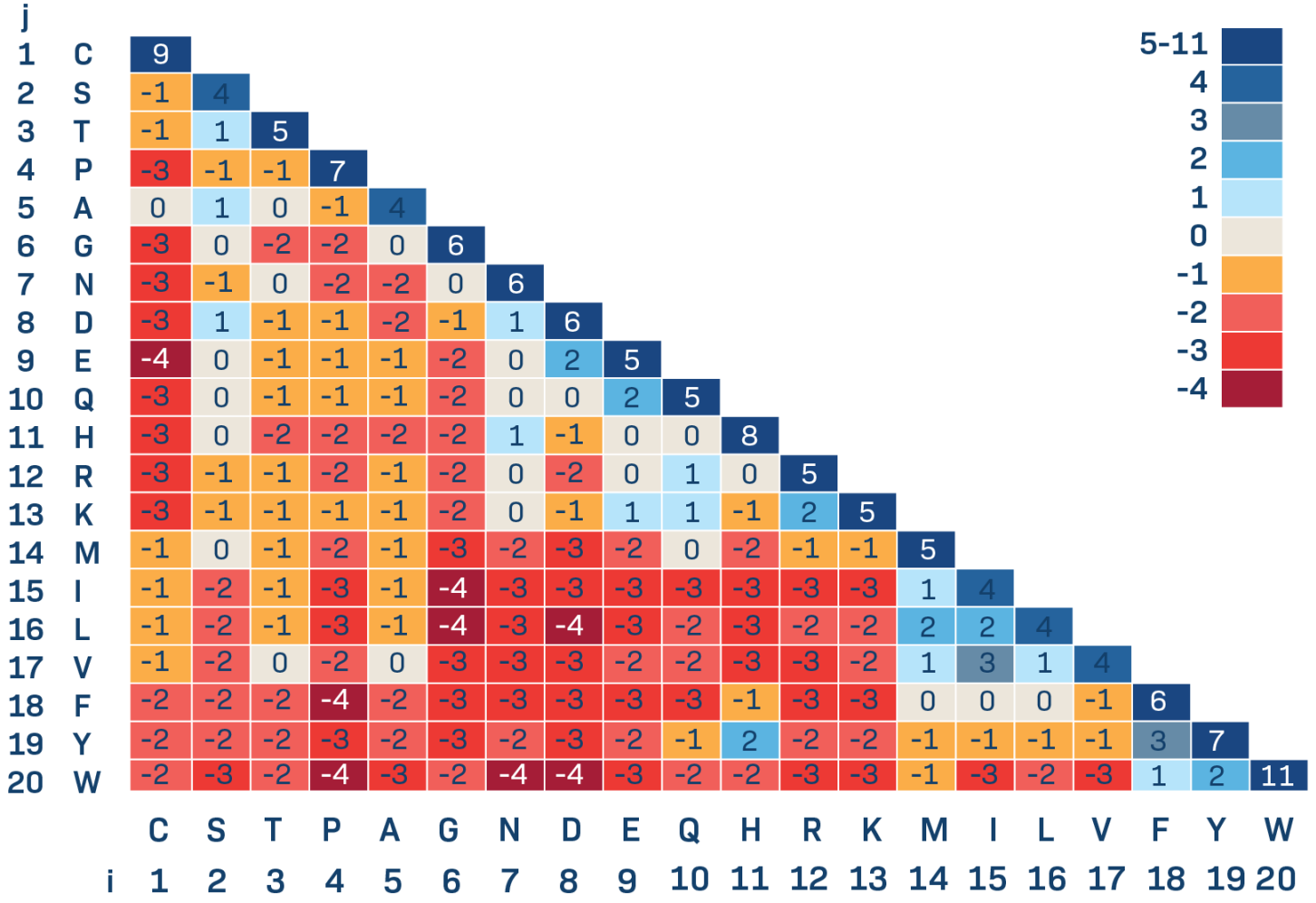
Scaling factor used to produce scores that can be rounded to integers; set to 0.5 in H&H '92.

A Family of Matrices

Henikoff and Henikoff divided the family clusters into sub-clusters by their percentage of similarity.

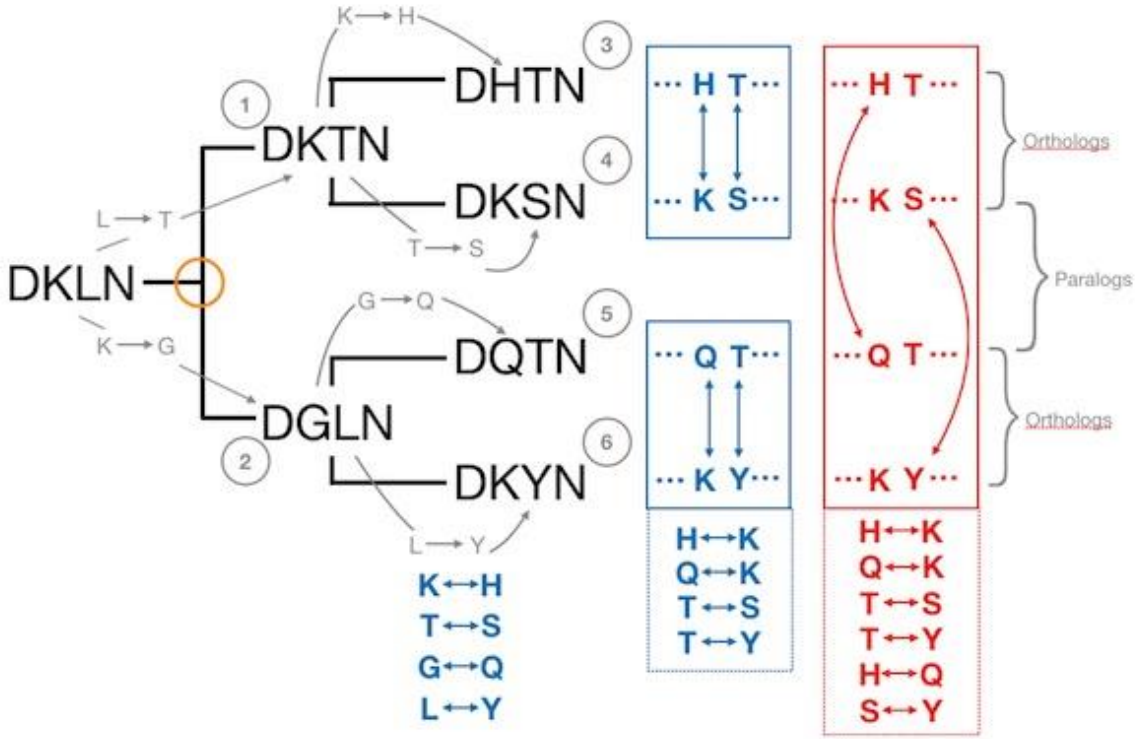
This division resulted in BLOSUM family of matrices where the associated number, e.g., BLOSUM65 means scores are from a cluster of sequences where sequences are at least 65% similar, in BLOSUM80 matrix scores are from clusters with at least 80% similarity and so on.

BLOSUM62


j
1 C 9
2 S -1 4
3 T -1 1 5
4 P -3 -1 -1 7
5 A 0 1 0 -1 4
6 G -3 0 -2 -2 0 6
7 N -3 -1 0 -2 -2 0 6
8 D -3 1 -1 -1 -2 -1 1 6
9 E -4 0 -1 -1 -1 -2 0 2 5
10 Q -3 0 -1 -1 -1 -2 0 0 2 5
11 H -3 0 -2 -2 -2 -2 1 -1 0 0 8
12 R -3 -1 -1 -2 -1 -2 0 -2 0 1 0 5
13 K -3 -1 -1 -1 -1 -2 0 -1 1 1 -1 2 5
14 M -1 0 -1 -2 -1 -3 -2 -3 -2 0 -2 -1 -1 5
15 I -1 -2 -1 -3 -1 -4 -3 -3 -3 -3 -3 -3 -3 1 4
16 L -1 -2 -1 -3 -1 -4 -3 -4 -3 -2 -3 -2 -2 2 2 4
17 V -1 -2 0 -2 0 -3 -3 -3 -2 -2 -3 -3 -2 1 3 1 4
18 F -2 -2 -2 -4 -2 -3 -3 -3 -3 -3 -1 -3 -3 0 0 0 -1 6
19 Y -2 -2 -2 -3 -2 -3 -2 -3 -2 -1 2 -2 -2 -1 -1 -1 -1 3 7
20 W -2 -3 -2 -4 -3 -2 -4 -4 -3 -2 -2 -3 -3 -1 -3 -2 -3 1 2 11
C S T P A G N D E Q H R K M I L V F Y W
i 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
5-11
4
3
2
1
0
-1
-2
-3
-4

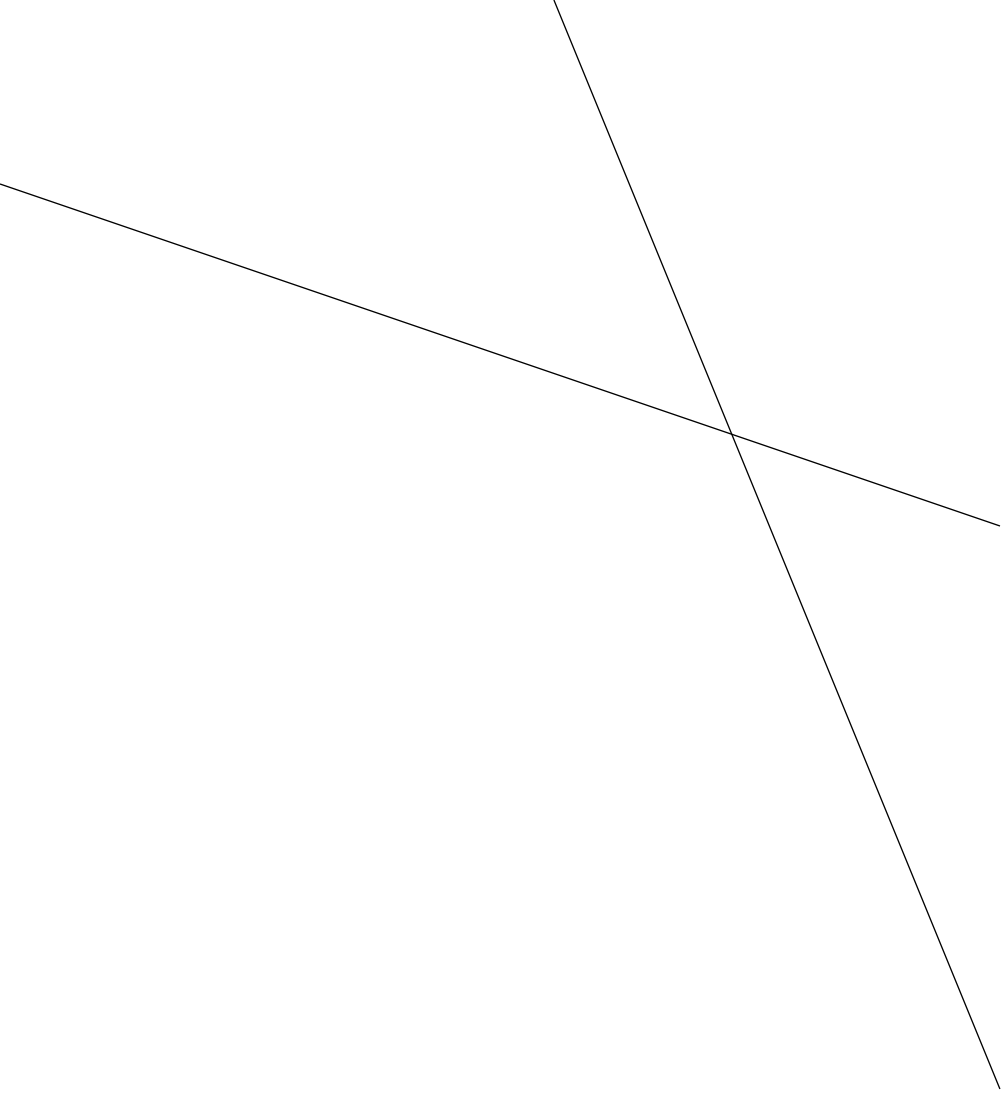
PAM Matrices

Accepted Point Mutation


K → H
DHTN
3
1
DKTN
4
DKSN
L → T
T → S
DKLN
G → Q
5
DQTN
K → G
DGLN
6
2
DKYN
L → Y
K↔H
T↔S
G↔Q
L↔Y
··· H T ···
··· K S ···
··· Q T ···
··· K Y ···
H↔K
Q↔K
T↔S
T↔Y
··· H T ···
··· K S ···
··· Q T ···
··· K Y ···
H↔K
Q↔K
T↔S
T↔Y
H↔Q
S↔Y
Orthologs
Paralogs
Orthologs

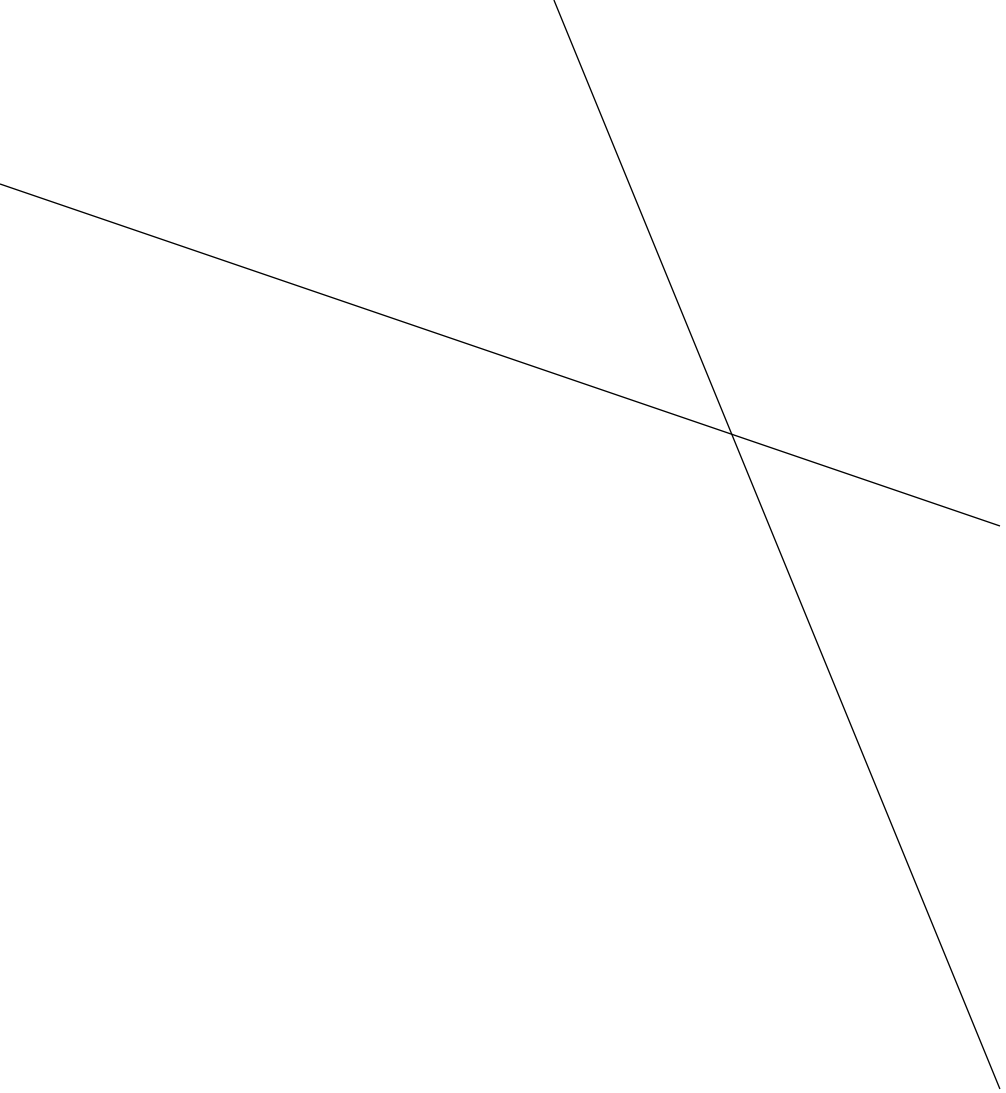
More information:

https://bioinformaticshome.com/bioinformatics_tutorials/sequence_alignment/substitution_matrices.html



How to select the right substitution matrix?

Statistical Significance of Alignment

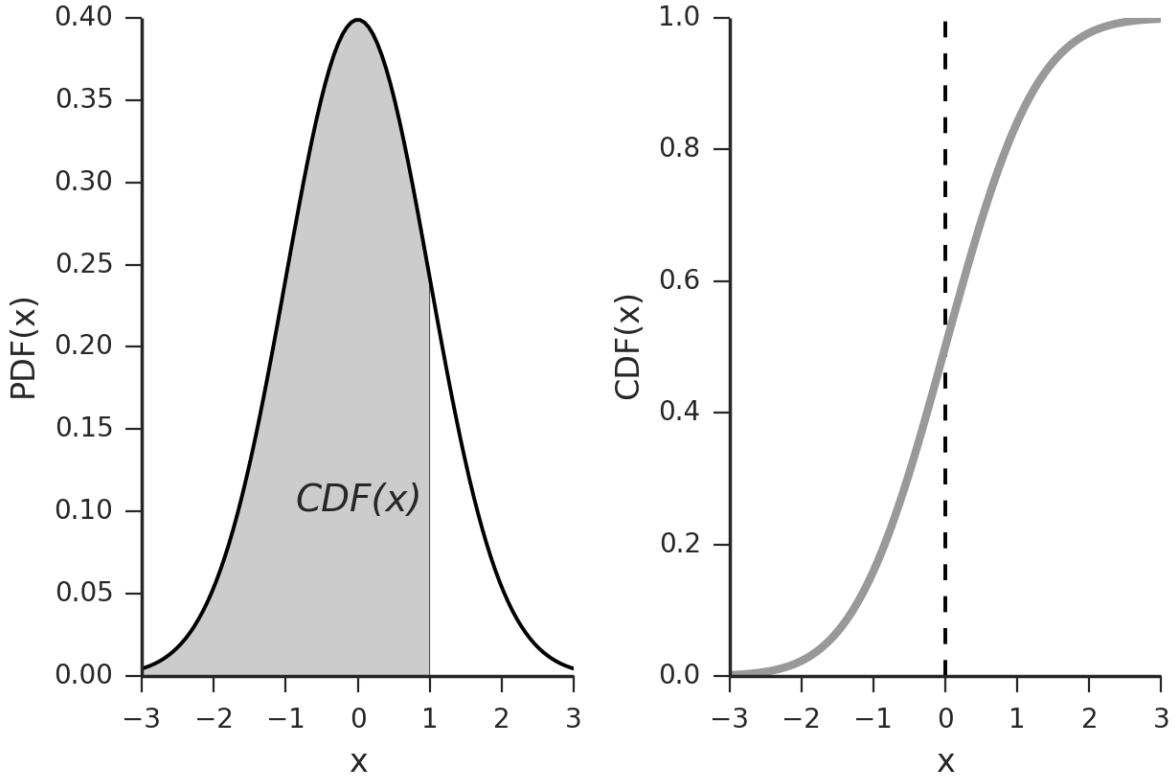


Database Searching

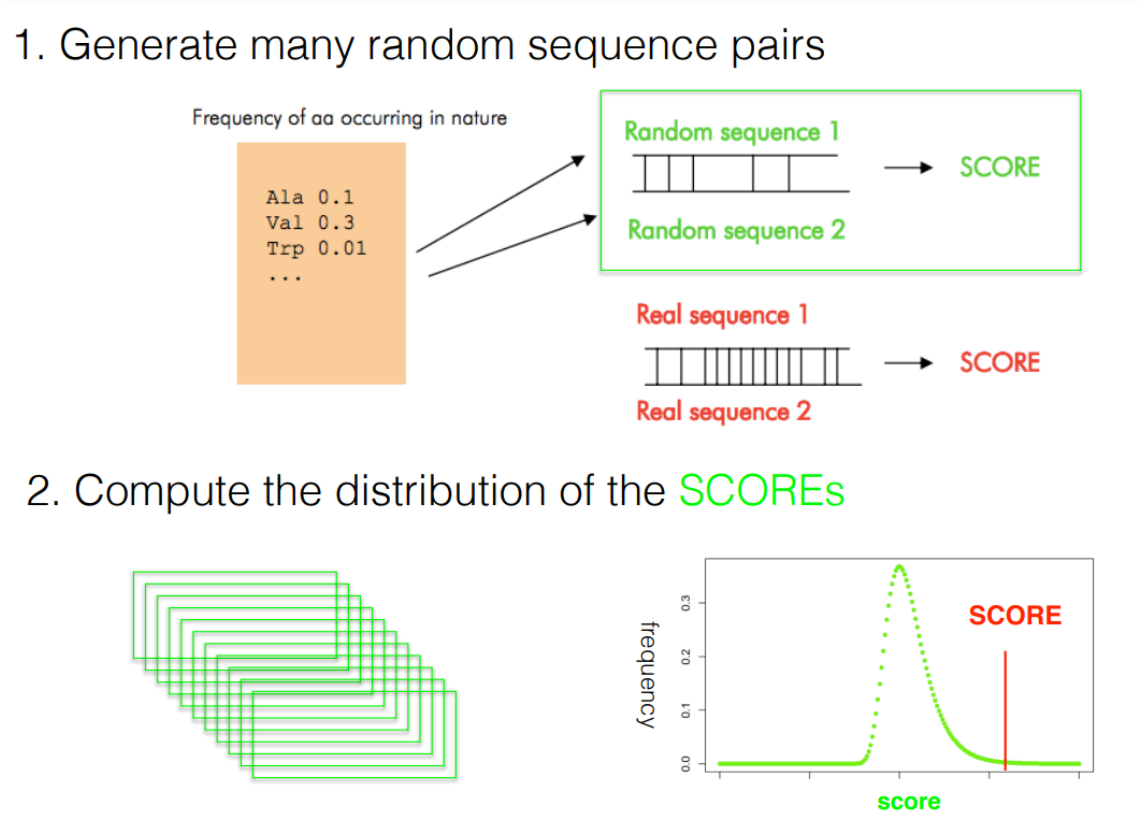
BLAST

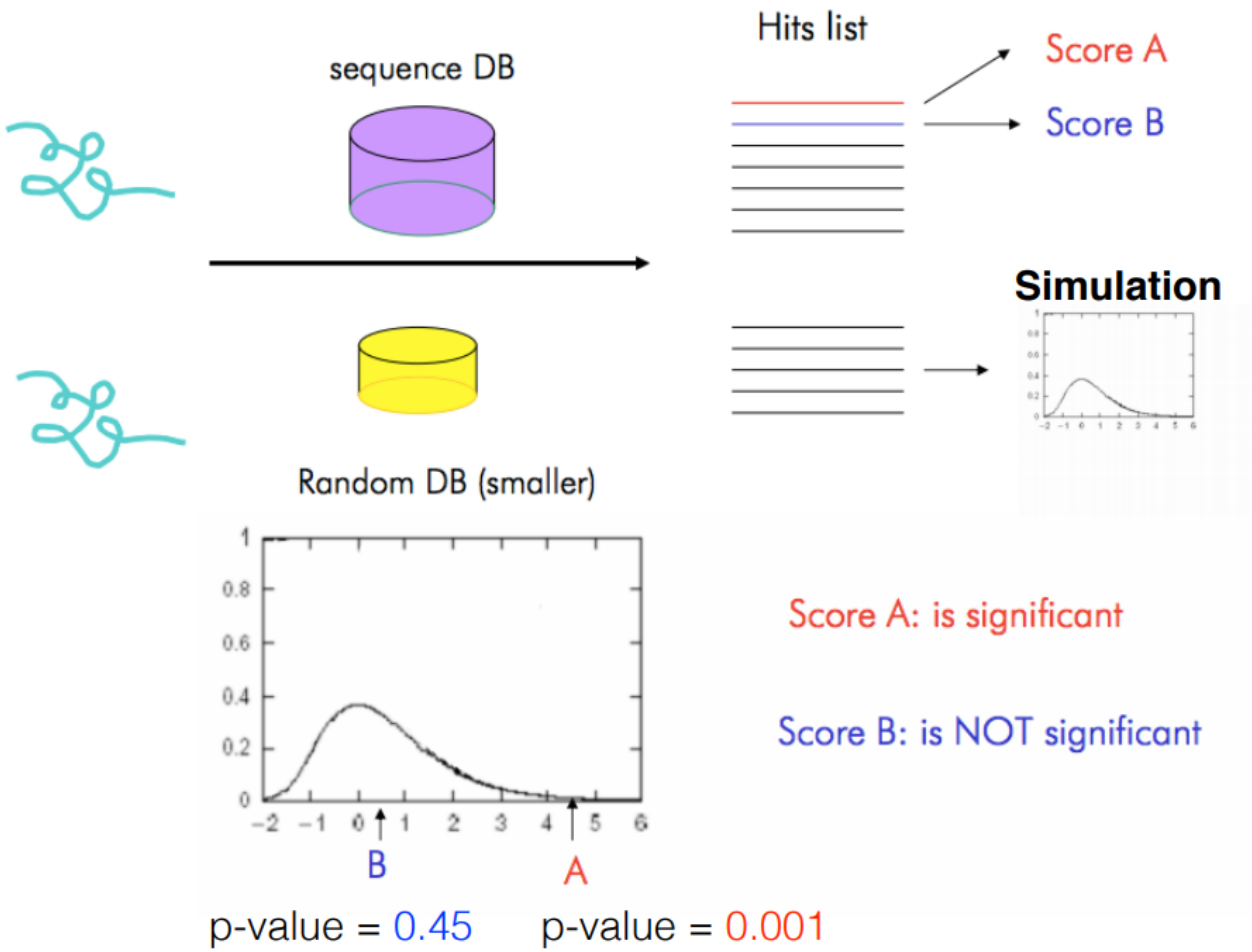
Given two random sequences of lengths m and n:
What is the probability that they will produce an MSP score of >= S ?

Given two random sequences of lengths m and n:
What is the probability that they will produce an MSP score of >= S ?

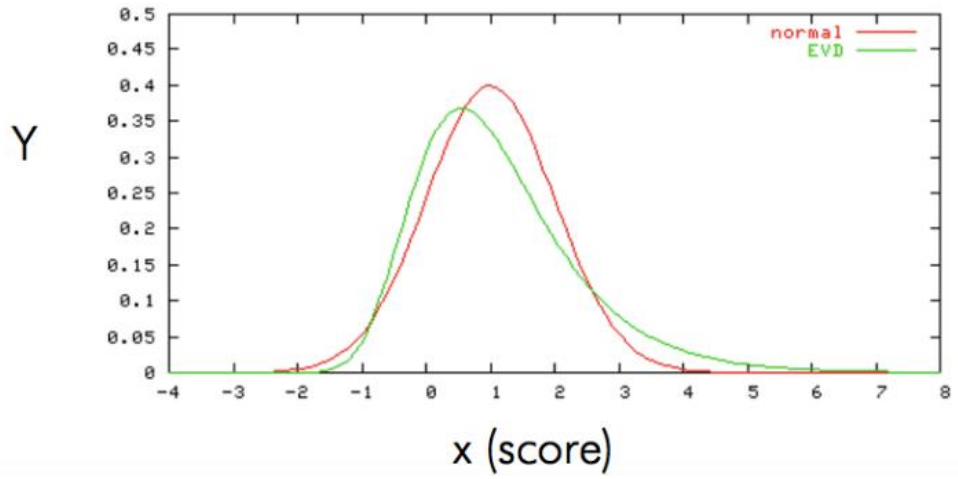


Given two random sequences of lengths m and n:
What is the probability that they will produce an MSP score of >= S ?

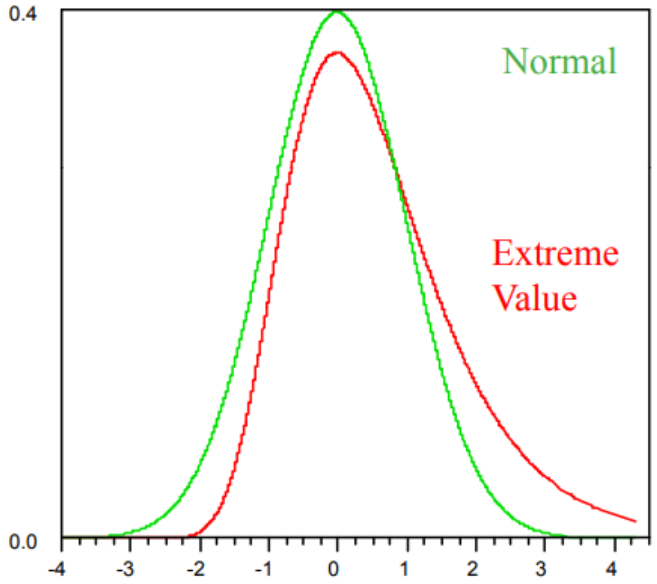



sequence DB
Hits list
Score A
Score B
Simulation
Random DB (smaller)
1
0.8
0.6
0.4
0.2
0
−2 −1 0 1 2 3 4 5 6
B
A
Score A: is significant
Score B: is NOT significant
p-value = 0.45
p-value = 0.001

Karlin and Altschul observed that in the framework of local alignments without gaps: the distribution of random sequence alignment scores follow an EVD.



Normal vs. Extreme Value Distribution



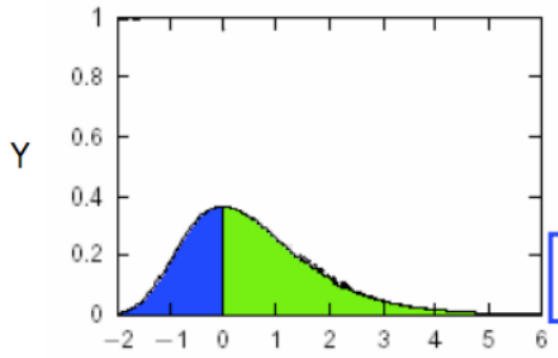
Normal distribution:

$y = (1/\sqrt{2\pi})e^{-x^2/2}$

Extreme value distribution:

$y = e^{-x - e^{-x}}$

Extreme value distribution



$$Y = \lambda \exp[-\lambda(x-\mu) - e^{-\lambda(x-\mu)}]$$

$$\int$$

$$P(Z < x) = \exp[-e^{-\lambda(x-\mu)}]$$

$$P(Z \geq x) = 1 - \exp[-e^{-\lambda(x-\mu)}]$$

P-value = the probability of obtaining a score equal or greater than x by chance

In a database of size N: P x N = E

- P-value:

Probability that an alignment with this score occurs by chance in a database of size N.
The closer the P-value is towards 0, the better the alignment

- E-value:

Number of matches with this score one can expect to find by chance in a database of size N.
The closer the E-value is towards 0, the better the alignment

→ Smaller E-value, more significant in statistics
Bigger E-value , by chance

Parameters

$$P(Z \geq x) = 1 - exp[-e^{-\lambda(x-\mu)}] \qquad (1)$$

μ, λ : parameters depend on the length and composition of the sequences and on the scoring system: μ is the mode (highest point) of the distribution and λ is the decay parameter
-They can me estimated by making many alignments of random or shuffled sequences.
- For alignments without gaps they can be calculated from the scoring matrix and then :

$$P(Z \geq x) = 1 - exp[-Kmne^{-\lambda x}] \qquad (2)$$

K: is a constant that depend on the scoring matrix values and the frequencies of the different residues in the sequences.
m,n : sequence lengths

Approximation:

if x is very small, then 1-exp(-x) can be approximated by x

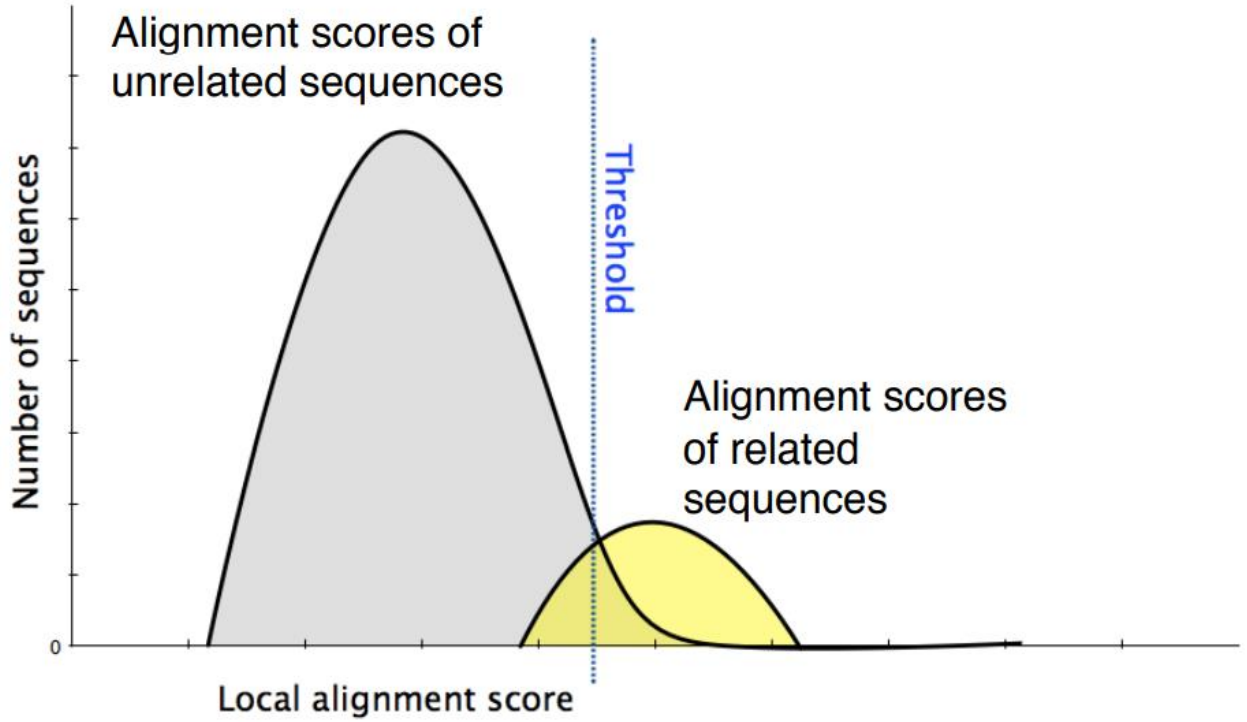
Therefore,

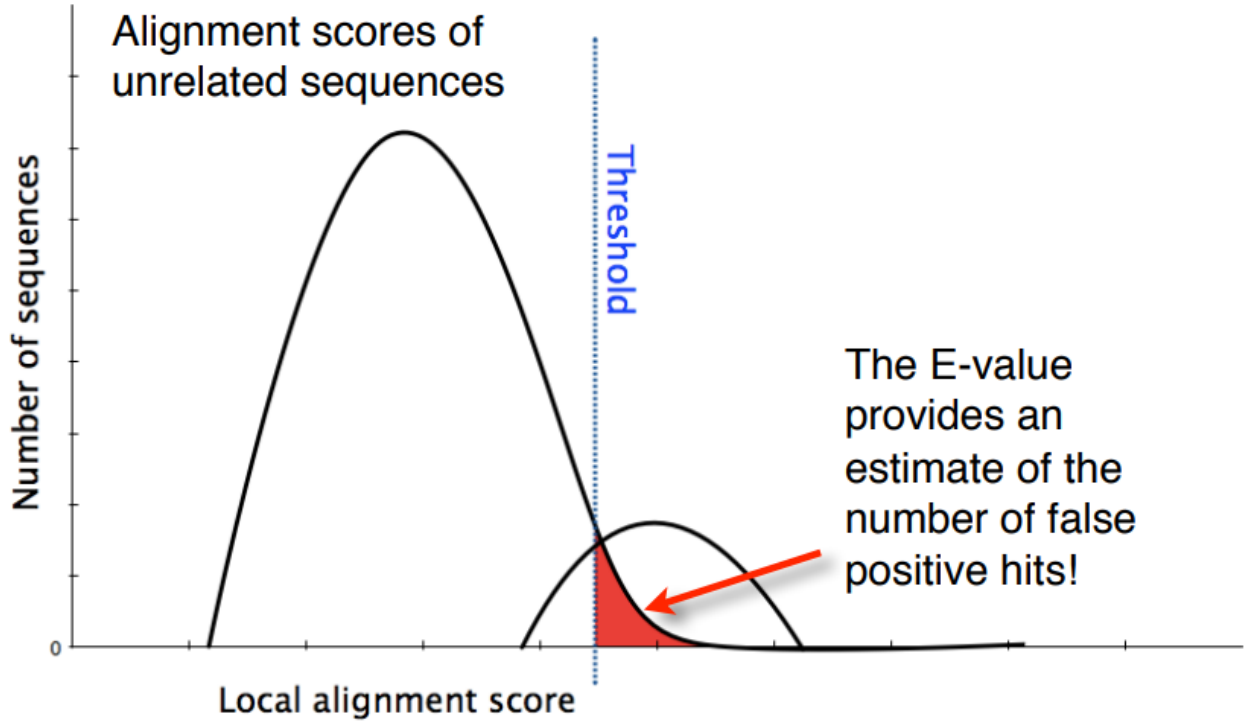
$$P(Z>=x) \sim e^{-\lambda(x-\mu)} = Kmn\, e^{-\lambda x}$$

So E-value = DatabaseLength * p-value

$$E\text{-value} = KNme^{-\lambda x}$$

where N is the database size (not the aligned length n)

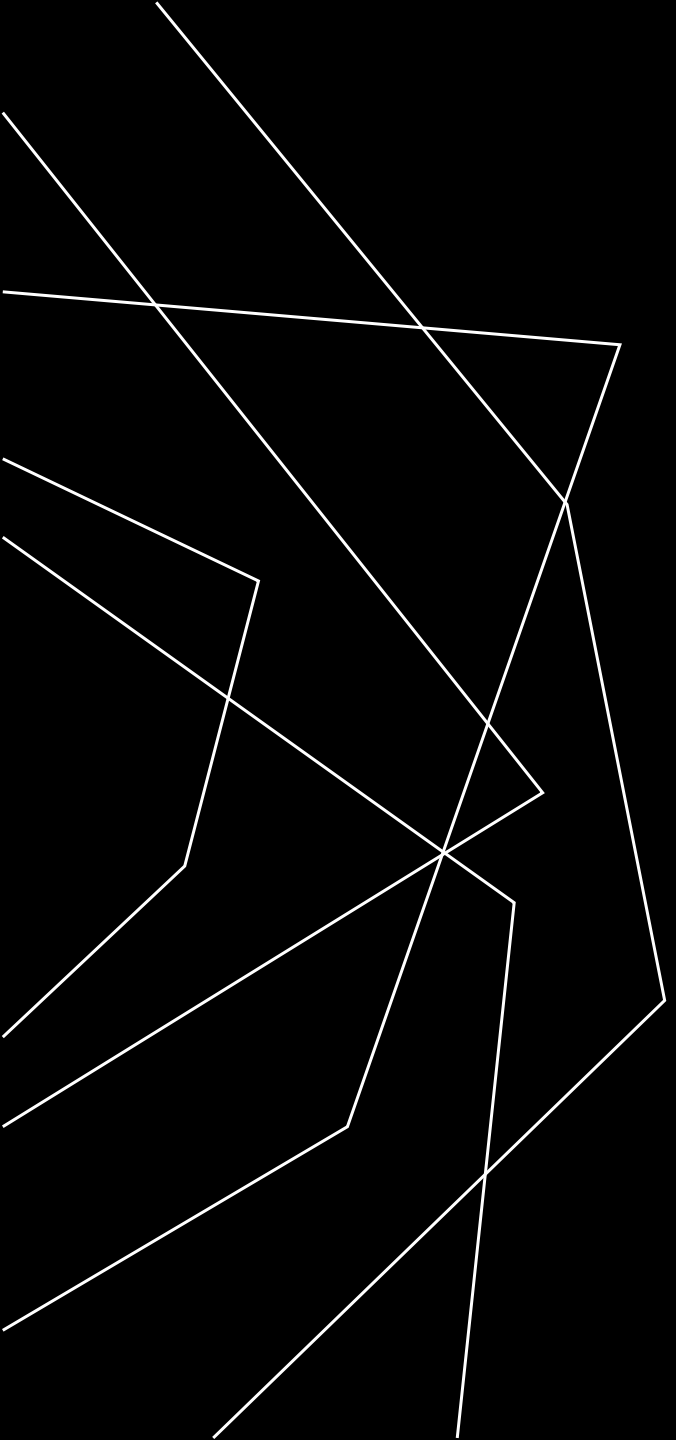

Alignment scores of
unrelated sequences
Threshold
Alignment scores
of related
sequences
Number of sequences
0
Local alignment score



The E value is the expected number of hits that are as good or better than the observed local alignment score (with this score or better) if the query and database are random with respect to each other.

$$E(S \geq x) = Kmne^{-\lambda x}$$

- $K < 1$ is a proportionality constant that corrects the mn "space factor" for the fact that there are not really mn independent places that could have produced score $S \geq x$.
- K has little effect on the statistical significance of a similarity score
- λ is closely related to the scoring matrix used and it takes into account that the scoring matrices do not contain actual probabilities of co-occurence, but instead a scaled version of those values. To understand how λ is computed, we have to look at the construction of scoring matrices.



THANK YOU

Q&A