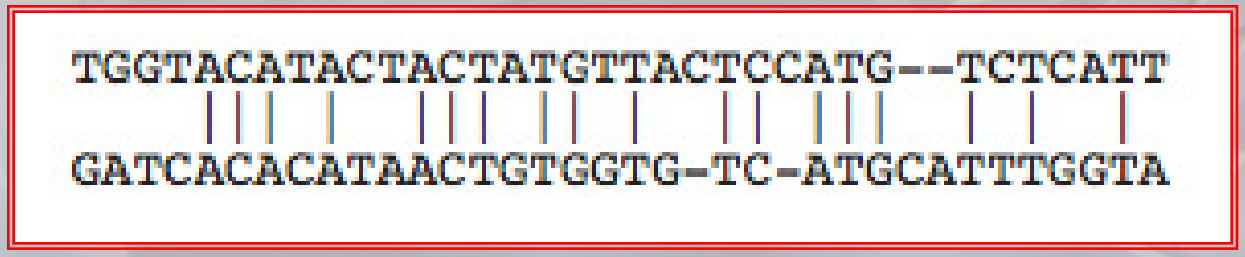
Database search and pairwise alignments

"It is a capital mistake to theorize before one has data. Insensibly one begins to twist facts to suit theories, instead of theories to suit facts."
(A. Conan Doyle, *A scandal in Bohemia*, Strand Magazine, July 1891)

Table of contents

Introduction – 1

- Each alignment between two or more nucleotide or amino acid sequences is an explicit assumption about their common evolutionary history
  - Comparisons among related sequences have facilitated many advances in understanding their information content and their function
  - Techniques for sequence alignment and sequence comparison, and similarity search algorithms in biological databases, are fundamental in Bioinformatics



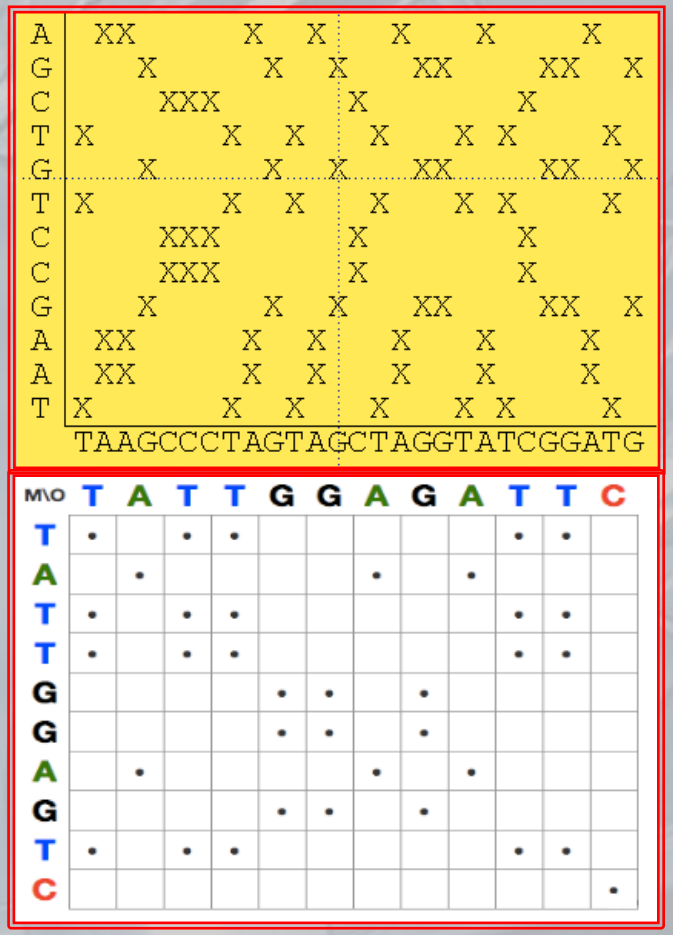
Introduction – 2

- Sequences closely related to each other are usually easy to align and, conversely, the quality of an alignment is an important indicator of their level of correlation
- Sequence alignments are used to:
  - Determine the function of a newly discovered genetic sequence (comparison with similar sequences)
  - Determine the evolutionary relationships between genes, proteins, and entire species
  - Predict the structure and the function of new proteins based on known "similar" proteins

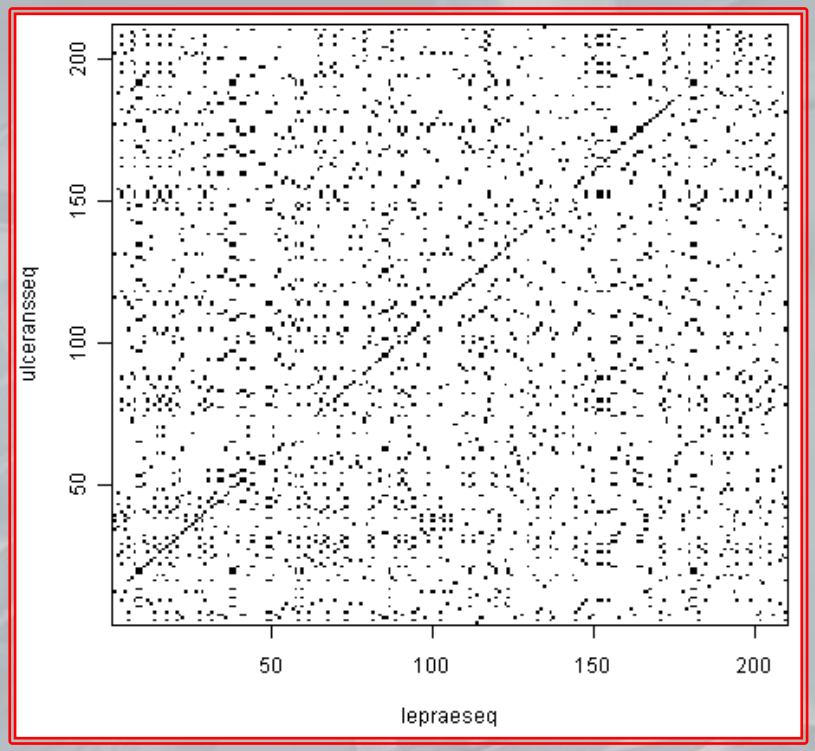
Dot plots – 1

- Probably, the simplest method to reveal analogies between two sequences consists in displaying their similarity regions using dot plots
  - The dot plot is a graphical method to display pairwise similarities
  - Less intuitive is its close relationship with pairwise alignments
- The dot plot is represented by a table or a matrix or, alternatively, in a Cartesian plane
  - The rows or the y–axis correspond to the elements of a sequence, and the columns or the x–axis to the elements of the other
  - The dots are posed in each position where the elements of both sequences coincide

Dot plots – 2



(a)

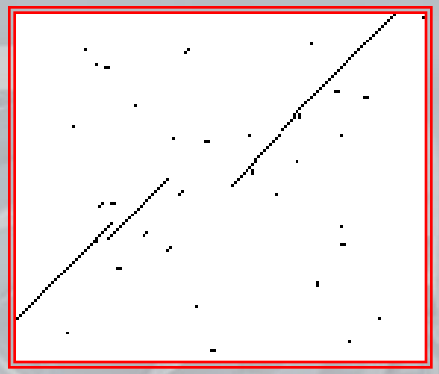


(b)

Dot plots: (a) matrix representations and (b) graphical representation in the Cartesian plane

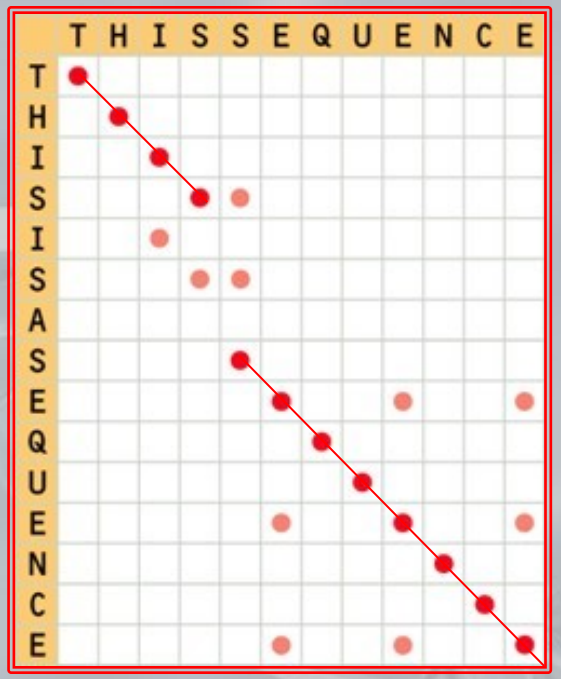
Dot plots – 3

- The similarity regions will thus be viewed as diagonal lines, that proceed from South–West to North–East; repeated sequences will produce parallel diagonals
- Therefore, dot plots capture, in a single image, not only the overall similarity between two sequences, but also the complete set and the relative quality of the different possible alignments
- Often, some similarity may be shifted, so as to appear on parallel, but not collinear, diagonals
  - This indicates the presence of insertion/deletion phenomena occurred in the segments between the similarity regions



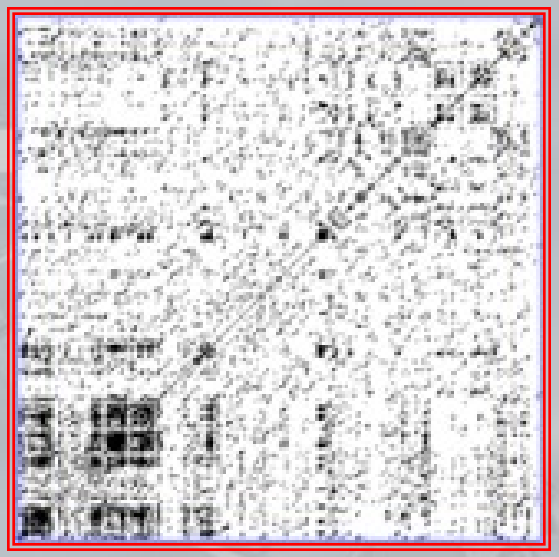
Dot plots – 4

- Example
  - Diagonal lines represent alignments
  - Horizontal lines between aligned sequences indicate that "something is lost" in one sequence
  - The longest alignments are "This" and "sequence", while "is a" is lost in the horizontal sequence
  - The pink dots represent noise (i.e., spurious alignments)



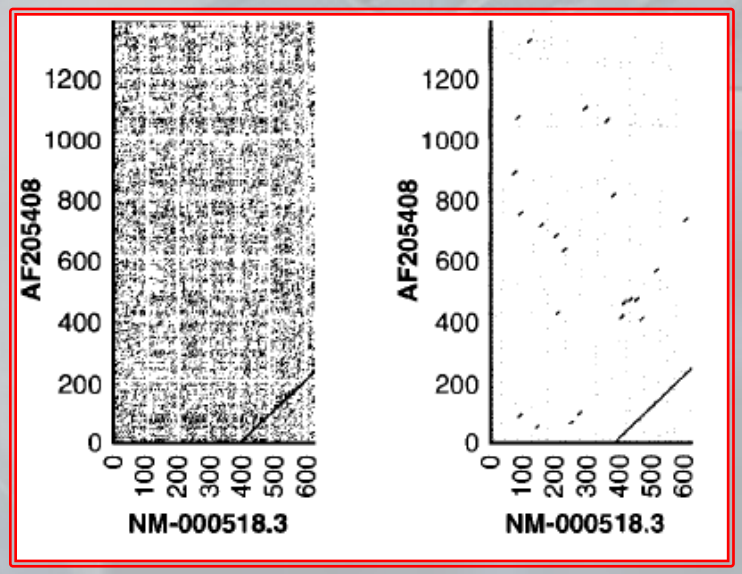
Dot plots – 5

- Indeed, in the dot plot matrices, random identities produce a high background noise (especially for long sequences)
- This happens almost always in the alignments between nucleic acids, due to the alphabet composed by only four letters
  - To reduce the noise, short sequences (in sliding windows) should be compared instead of single nucleotides
  - In this case, the dot is reported only when s nucleotides coincide within a window of dimension w



Dot plots – 6

- Increasing the s value corresponds to increase the requested precision (maximum for $s = w$)
- Obviously, the variation of w and s has a significant influence on the background noise
- The best experimental values for w and s, with respect to nucleotide and protein sequences, are empirically determined by a *trial–and–error* procedure



A complete dot plot comparing nucleotide sequences from the β–globin genes of human and orangutan (w=10, s=8)

Dot plots – 7

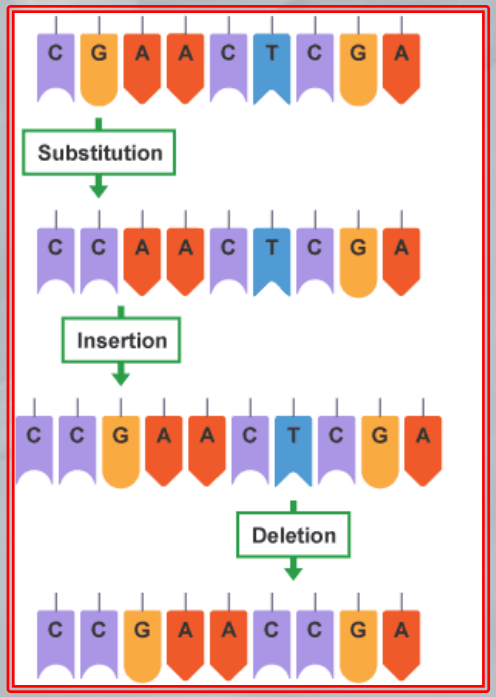
- Particularly in the case of proteins, a dot plot matrix, that actually considers only identities, does not provide a true indication of the similarity relations between sequences, since the non–identity among amino acids can have very different biological implications
- In fact:
  - In some situations, the replacement of a residue with a different one, but with very similar properties (e.g.: leucine and isoleucine), can be almost irrelevant
  - In other cases, two non–identical residues can have very different properties

Simple alignments – 1

- A simple pairwise alignment consists in matching pairs of characters belonging to two sequences
- The alignment of nucleotide or amino acid sequences reflects their evolutionary relationship, namely their homology, i.e., the presence of a common ancestor
  - A score for homology does not exist: At any given position of an alignment, the two sequences may share an ancestor character or not
  - The overall similarity can instead be quantified by means of a fraction

Simple alignments – 2

- In particular, in any given position within a sequence, three types of changes may occur:
  - A substitution that replaces one character with another
  - An insertion, which adds one or more characters
  - A deletion, which eliminates one or more characters



- In Nature, insertions and deletions are significantly less frequent than substitutions
- Since there are no homologues of nucleotides inserted or deleted, gaps are commonly added in the alignments, in order to reflect the occurrence of this type of changes

Simple alignments – 3

- In the simplest case, in which gaps are not allowed, the alignment of two sequences is reduced to the choice of the starting point for the shorter sequence

```
AATCTATA        AATCTATA        AATCTATA
AAGATA           AAGATA           AAGATA
```

- To determine which of the three alignments is "optimal", it is necessary to establish a score to comparatively evaluate them

$$\sum_{i=1}^{n} \begin{cases} \text{Correspondence score, if } seq1_i = seq2_i \\ \text{Non–correspondence score, if } seq1_i \neq seq2_i \end{cases}$$

where n is the length of the longest sequence

- For a score of mismatching/matching equal to 0/1, the three alignments are evaluated respectively 4, 1 and 3

Gaps

- Considering the possibility that insertion and deletion events can occur significantly increases the number of possible alignments between pairs of sequences
- For example, the two sequences **AATCTATA** and **AAGATA** that can be aligned without gaps in only three ways, admit 28 different alignments, with the insertion of two gaps within the shorter sequence
- Examples

AATCTATA	**AATCTATA**	**AATCTATA**
AAG–AT–A	**AA–G–ATA**	**AA––GATA**

Simple penalties for gap insertion

- Introduction, in the alignment evaluation score, of a penalty term for a gap insertion (gap penalty)

$$\sum_{i=1}^{n} \begin{cases} \text{Penalty for a gap insertion, if } seq1_i = \text{"}-\text{"} \text{ or } seq2_i = \text{"}-\text{"} \\ \text{Correspondence score, if } seq1_i = seq2_i \\ \text{Non}-\text{correspondence score, if } seq1_i \neq seq2_i \end{cases}$$

  - Assuming a score of mismatching/matching equal to 0/1 and a gap penalty equal to −1, the scores for the following three alignments with gaps (out of 28) would be 1, 3, 3

AATCTATA	**AATCTATA**	**AATCTATA**
AAG–AT–A	**AA–G–ATA**	**AA––GATA**

Penalties for the presence and the length of a gap – 1

- Using a simple gap penalty, it is common to evidence many "optimal" alignments (depending on the selected criterion)
  - Choose different penalty values for single gaps and gaps that appear in sequence
- Concretely, any pairwise alignment represents a hypothesis about the evolutionary path that two sequences have undertaken from the last common ancestor
- When several competing hypotheses are considered, the one that invokes the fewest unlikely events is, by definition, the most likely correct one

Penalties for the presence and the length of a gap – 2

- Let s_1 and s_2 be two arbitrary DNA sequences of length 12 and 9, respectively
  - Each alignment will necessarily have three gaps in the shorter sequence
  - Assuming that s_1 and s_2 are homologous sequences, the difference in length can be caused by the insertion of nucleotides in the longer sequence, or by the deletion of nucleotides in the shorter sequence, or by a combination of the two events
  - Since the sequence of the ancestor is unknown, no methods exist able to determine the cause of a gap, which is attributed generally to an indel event (insertion/deletion)
  - Moreover, since sequential insertions/deletions are not uncommon, it is statistically more likely that the difference in length between the two sequences was due to a single indel of three nucleotides, rather than to three distinct indels

Penalties for the presence and the length of a gap – 3

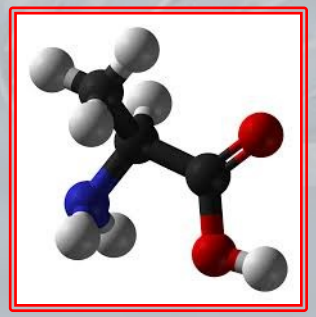
- The scoring function has to reward alignments that are most plausible from the evolutionary point of view
- By assigning a penalty on the length of the gap (which depends on the number of sequential characters missed) lower than the penalty for the creation of new gaps, the scoring function rewards the alignments that show sequential gaps
- Example: Using a gap creation penalty equal to –2, a length penalty of –1 (for each missed character), and mismatching/matching values equal to 0/1, the scores in the three cases below are respectively –3, –1, 1

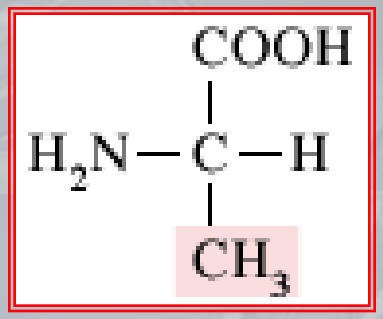
AATCTATA	**AATCTATA**	**AATCTATA**
AAG–AT–A	**AA–G–ATA**	**AA––GATA**

Score matrices – 1

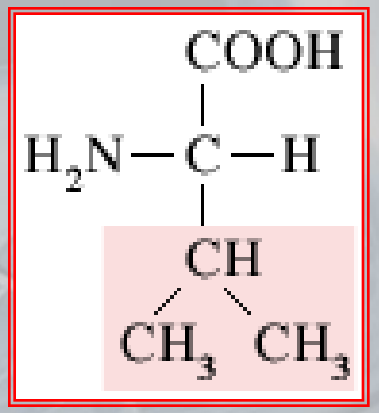
- Just as the gap penalty, that can be adapted to reward alignments evolutionarily more plausible, so the mis-matching penalty can be made non–uniform, based on the simple observation that some substitutions are more common (and less dangerous) than others
- Example: Let us consider a protein sequence, which has an alanine at a given position
  - A substitution with another small and hydrophobic amino acid, such as valine, has a lower impact on the resulting protein with respect to a replacement with a large and charged residue such as, for example, lysine

Score matrices – 2

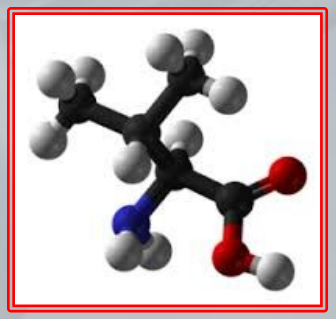


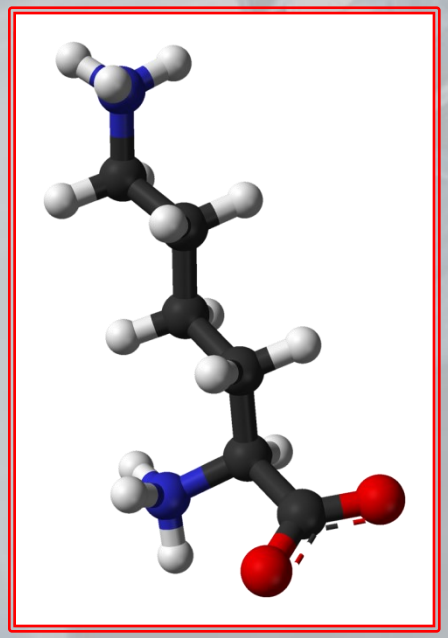


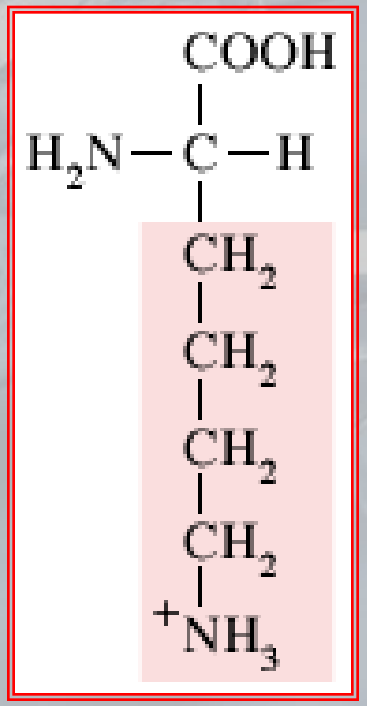
Alanine



Valine







Lysine

Score matrices – 3

- Intuitively, a conservative substitution, unlike a more drastic one, may occur more frequently, because it preserves the original functionality of the protein
- Given an alignment score for each possible pair of nucleotides or residues, the score matrix is used to assign a value to each position of an alignment, except gaps
- Example

Nucleotide matches are moderately rewarded, while a small penalty is given to transition events (substitution between purines or pyrimidines, **A**-**G**/**C**-**T**); instead, a more severe penalty is assigned for transversions, in which a purine replaces a pyrimidine or vice versa

	A	**T**	**C**	**G**
A	1	–5	–5	–1
T	–5	1	–1	–5
C	–5	–1	1	–5
G	–1	–5	–5	1

Score matrices – 4

- Several criteria can be considered in setting up a score matrix for amino acid sequence alignments
  - Physico–chemical similarity
  - Genetic similarity
  - Observed replacement frequencies
- In physico–chemical similarity–based matrices, the substitution between two amino acids, which, however, have both functional aromatic groups (hydrophobic amino acids, with a side chain containing a benzene ring, such as phenylalanine, tyrosine and tryptophan) could receive a positive score, while substituting non–polar with charged amino acids could be penalized

Score matrices – 5

- Score matrices can be derived according to the hydrophobicity, the presence of charge, the electronegativity, and the size of the involved residues
- Alternatively, similarity criteria based on the encoding genome can also be used: the assigned score is proportional to the minimum number of nucleotide substitutions necessary to convert a codon to another
  - Difficulty in combining, in a single "significant" matrix, chemical, physical and genetic scores

Score matrices – 6

- The most common method to derive score matrices consists in observing the actual frequencies of amino acid substitutions in Nature
- If a replacement which involves two particular amino acids is frequently observed, their alignment obtains a favorable score
- Vice versa, alignments between residues which, during evolution, are rarely observed must be penalized

PAM matrices – 1

- PAM matrices exploit the concept of *Point* – or *Percent* – *Accepted Mutations*; they were proposed in 1978, by M. Dayhoff *et al.*, on the basis of a study on molecular phylogeny involving 71 protein families
- PAM matrices were developed by examining mutations within superfamilies of closely related proteins, also noting how observed substitutions did not happen at random
  - Some amino acid substitutions occur more frequently than others, probably because they do not significantly alter the structure and the function of a protein
  - Homologous proteins need not to be necessarily constituted by the same amino acids in each position

PAM matrices – 2

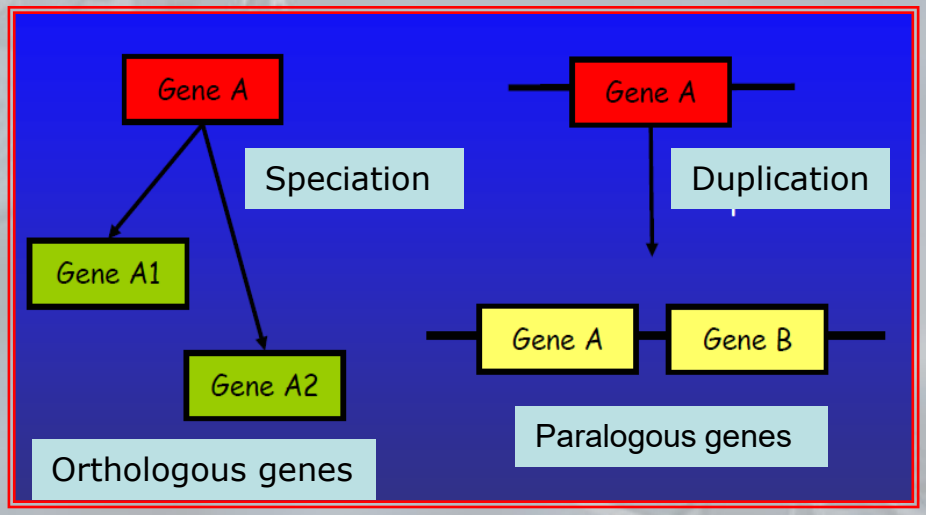
- Two proteins are "distant" 1 PAM unit if — on average — they differ for a single amino acid out of 100, and if the substitution is *accepted*, i.e., it does not result in a loss of functionality
  - Therefore… two sequences s_1 and s_2 are distant 1 PAM if s_1 can be transformed into s_2 with a point substitution per 100 amino acids, on average
  - Since the amino acid at a certain position may change several times and then may return to the original character, the two sequences that are 1 PAM may be the evolutionary product of a set of unobservable substitutions that involved more than one amino acid out of 100

PAM matrices – 3

- In other words, the elements of a PAM 1 matrix answer the following question
  - Suppose we have a polypeptide sequence S at time t, and observe the evolutionary changes in the sequence until 1% of all amino acid residues have undergone substitutions at the time $t+n$
  - Let us call S' the new sequence at time $t+n$
  - What is the probability that a residue j in S is replaced by a residue i in S'?
  - The answer to this question corresponds to the element P_{ij} of the PAM 1 matrix

PAM matrices – 4

- Protein families used for PAMs collect orthologous proteins (which perform the same function in different organisms); instead, pathological changes, that are associated to loss of functionality, are not included in this collection
- To generate a PAM 1 matrix, we consider pairs of very similar protein sequences (with an identity >85%), for which an alignment can be derived without ambiguity



PAM matrices – 5

- Based on this set of proteins, a PAM 1 score matrix can be defined as follows:
  - Calculation of an alignment among sequences with very high identity
  - For each pair of amino acids i and j, calculation of F_{ij}, the number of times that the amino acid j is replaced by i
  - For each amino acid j, evaluation of the relative mutability n_j (the number of substitution in which such amino acid is involved, appropriately normalized)
  - Evaluation of M_{ij}, the "mutation probability" for each amino acid pair $j \rightarrow i$
  - Finally, each PAM 1 element, P_{ij}, is evaluated by applying the logarithm to M_{ij}, previously normalized w.r.t. the frequency of the residue i
- PAM 1 is also called the *log–odds* matrix

PAM matrices – 6

- Example (to be continued)

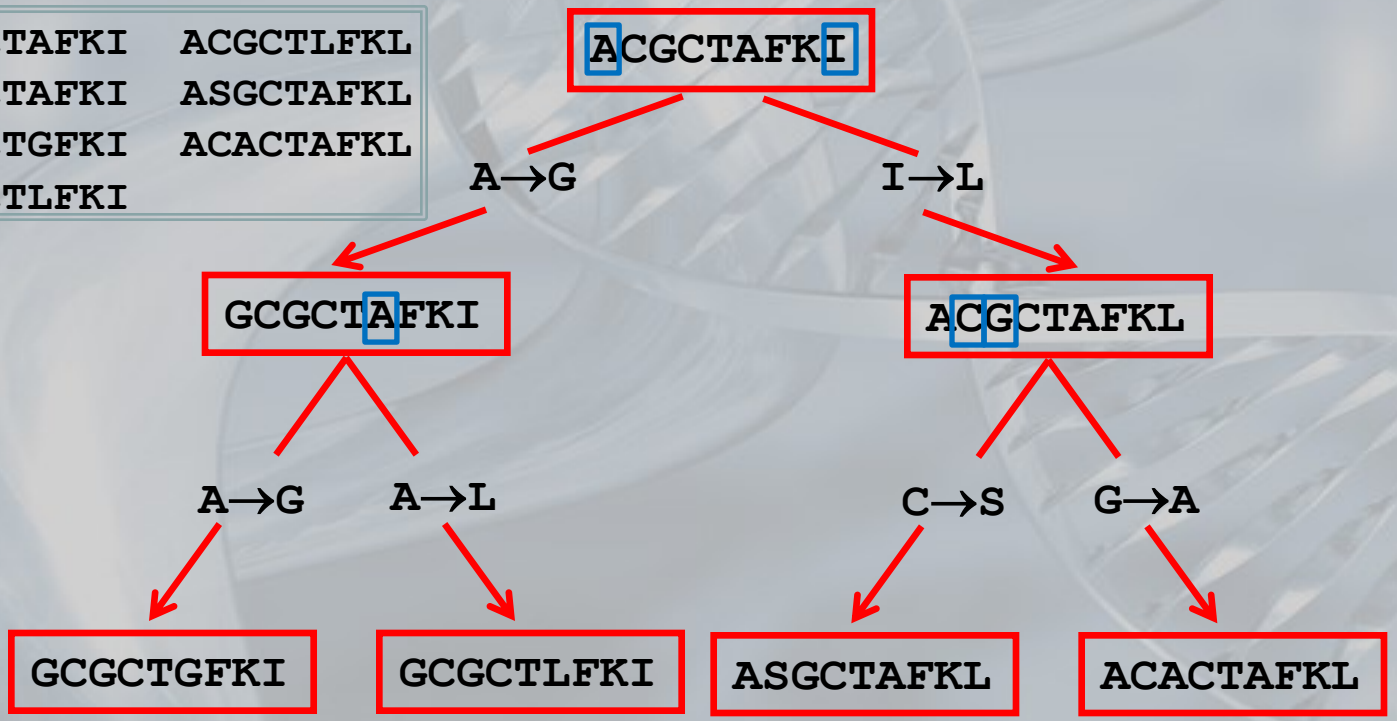
1) Construction of a multiple sequence alignment:

```
ACGCTAFKI        ACGCTLFKL
GCGCTAFKI        ASGCTAFKL
GCGCTGFKI        ACACTAFKL
GCGCTLFKI
```

PAM matrices – 7

- Example (to be continued)
  2) A phylogenetic tree is created, that indicates the order in which substitutions may have been occurred during evolution

ACGCTAFKI	ACGCTLFKL
GCGCTAFKI	ASGCTAFKL
GCGCTGFKI	ACACTAFKL
GCGCTLFKI	

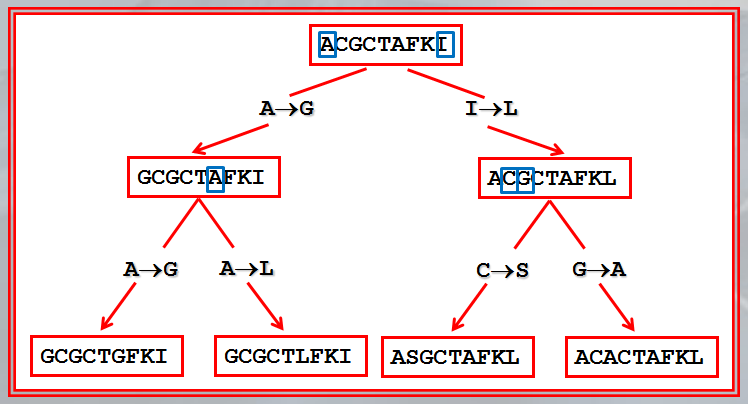


PAM matrices – 8

- Example (to be continued)

3) For each amino acid, we calculate the number of replacements with respect to any other amino acid
   - It is assumed that the substitutions are symmetric, that is they occur with the same probability with respect to a given pair of amino acids
   - For instance, in order to determine the substitution frequency between **A** and **G** (alanine and glycine), $F_{G,A}=F_{A,G}$, we count all the branches **A→G** and **G→A**

$F_{G,A}= 3$



PAM matrices – 9

- Example (to be continued)

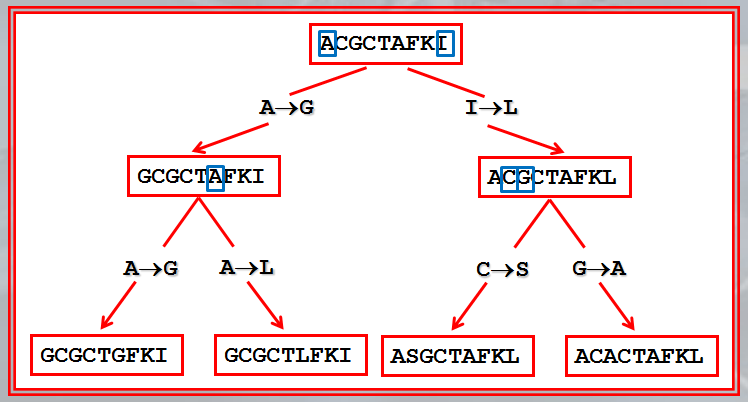
4) Calculation of the relative mutability n_j of each amino acid

- The mutability m_j is the number of times an amino acid is replaced by any other in the phylogenetic tree
- This number is then normalized by the total number of mutations that may have some effects on the alignment
- ...that is, the denominator of the fraction is given by the total number of substitutions in the tree, multiplied by two, multiplied by the frequency of the particular amino acid, multiplied by a scale factor equal to 100
- The scale factor 100 is used because the PAM 1 matrix represents the "substitution probability" per 100 residues

PAM matrices – 10

- Example (to be continued)
  - Let us consider the amino acid **A**: There are 4 substitutions involving **A** in the phylogenetic tree ($m_{\mathbf{A}}$= 4)
  - This value must be divided by twice the total number of substitutions (6×2=12), multiplied by the relative frequency of the residue $f_{\mathbf{A}}$ (10/63=0.159), multiplied by 100

$$n_{\mathbf{A}}=4/(12\times0.159\times100)=0.0209$$



PAM matrices – 11

- Example (to be continued)

5) Let us calculate the "mutation probability" M_{ij}, for each pair of amino acids

relative mutability of j

$$M_{ij}=n_j F_{ij}/\Sigma_k F_{kj}$$

total replacements involving j

total replacements $j \leftrightarrow i$

and then…

$$M_{\mathtt{G},\mathtt{A}}=(0.0209\times 3)/4=0.0156$$

where the denominator $\Sigma_k F_{kj}$ represents the total number of substitutions that involves **A** in the phylogenetic tree

PAM matrices – 12

- Example

6) Finally, each M_{ij} must be divided by the frequency of the residue i; the logarithm of the resulting value constitutes the corresponding element of the PAM matrix, P_{ij}
   - For **G**, the frequency $f_{\mathbf{G}}$ is equal to 0.159 (10/63)
   - For **G** and **A**, $P_{\mathbf{G},\mathbf{A}}=log(0.0156/0.159)\approx-1.01$

7) By repeating the above procedure for each pair of amino acids we can obtain all the extra–diagonal values of the PAM matrix, whereas P_{ii} are calculated posing

$$M_{ii}=1-\Sigma_{k\neq i}M_{ki}$$

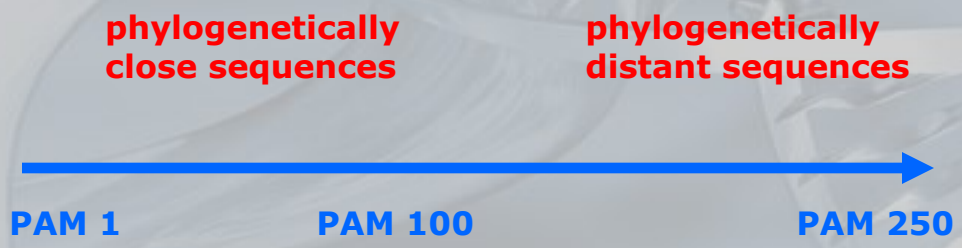
and executing 6)

PAM matrices – 13

- High order PAM matrices are generated by successive multiplications of the PAM 1 matrix, since the probability of two independent events is equal to the product of the probabilities of each individual event
- While for the PAM 1 matrix it holds that a mutational event corresponds to a difference of ≈1%, this is not true for higher order PAM matrices
- Indeed, subsequent mutations have a gradually increasing chance to happen in correspondence of already mutated amino acids
- The degree of difference increases with the increase in the number of mutations, but while the number of mutations can tend to infinity, the difference tends asymptotically to 100%

PAM matrices – 14

- The choice of the most suitable PAM matrix with respect to a particular alignment of sequences, depends on their length and on their correlation degree
- PAM 2 is calculated from PAM 1 assuming another evolutionary step
- PAM n is obtained from PAM n–1
- PAM 100, therefore, represents 100 evolutionary steps, in each of which there was a 1% of substitutions more compared to the previous step



- The most used matrix, in practice, is PAM250, which stands for an overall change of 250%; at this level, however, the amino acid sequences still retain a 40% similarity

PAM matrices – 15

		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	Ala	.18																			
R	Arg	-.15	.61																		
N	Asn	.02	0	.20																	
D	Asp	.03	-.13	.21	.39																
C	Cys	-.20	-.36	-.36	-.51	1.19															
Q	Gln	-.04	.13	.08	.16	-.54	.40														
E	Glu	.03	-.11	.14	.34	-.53	.25	.38													
G	Gly	.13	-.26	.03	.06	-.34	-.53	.25	.38												
H	His	-.14	.16	.16	.07	-.34	.29	.07	-.21	.65											
I	Ile	-.05	-.20	-.18	-.24	-.23	-.20	-.20	-.26	-.24	.45										
L	Leu	-.19	-.30	-.29	-.40	-.60	-.18	-.34	-.41	-.21	.24	.59									
K	Lys	-.12	.34	.10	.01	-.54	.07	-.01	-.17	0	-.19	-.29	.47								
M	Met	-.11	-.04	-.17	-.26	-.52	-.10	-.21	-.28	-.21	.22	.37	.04	.64							
F	Phe	-.35	-.45	-.35	-.56	-.43	-.47	-.54	-.48	-.18	.10	.18	-.53	.02	.91						
P	Pro	.11	-.02	-.05	-.10	-.28	.02	-.06	-.05	-.02	-.20	-.25	-.11	-.21	-.46	.59					
S	Ser	.11	-.03	.07	.03	0	-.05	0	.11	-.08	-.14	-.28	-.02	-.16	-.32	.09	.16				
T	Thr	.12	-.09	.04	-.01	-.22	-.08	-.04	0	-.13	.01	-.17	0	-.06	-.31	.03	.13	.26			
W	Trp	-.58	.22	-.42	-.68	-.78	-.48	-.70	-.70	-.28	-.51	-.18	-.35	-.42	.04	-.56	-.25	-.52	1.73		
Y	Tyr	-.35	-.42	-.21	-.43	.03	-.40	-.43	-.52	-.01	-.09	-.09	-.44	-.24	.70	-.49	-.28	-.27	-.02	1.01	
V	Val	.02	-.25	-.17	-.21	-.19	-.19	-.18	-.14	-.22	.37	.19	-.24	.18	-.12	-.12	-.10	.03	-.62	-.25	.43
		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
		Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val

Dayhoff's PAM 250 matrix

PAM matrices – 16

- It is worth noting that:
  - Each PAM matrix element P_{ij} describes how much the substitution of the amino acid A_j with the amino acid A_i is more (or less) frequent than a random mutation
  - Therefore:

    $P_{ij} > 0$ more frequent than a random mutation

    $P_{ij} = 0$ frequent as a random mutation

    $P_{ij} < 0$ less frequent than a random mutation

BLOSUM matrices – 1

- BLOSUM matrices (*BLOcks amino acid SUbstitution Matrices*) were introduced in 1992 by S. Henikoff and J. G. Henikoff to assign a score to substitutions between amino acid sequences
- Their purpose was to replace the PAM matrices, making use of the increased amount of data that had become available after the work of Dayhoff
- They were supposed to work better than PAMs especially with respect to poorly correlated sequences
- The BLOCKS database contains multiply aligned ungapped segments corresponding to the most highly conserved regions of proteins
- Each alignment block contains sequences with a number of identical amino acids greater than a certain percentage N
- From each block, it is possible to derive the relative frequency of amino acid replacements, which can be used to calculate a score matrix

BLOSUM matrices – 2

- The elements of the BLOSUM matrix, B_{ij}, are evaluated based on the following relation

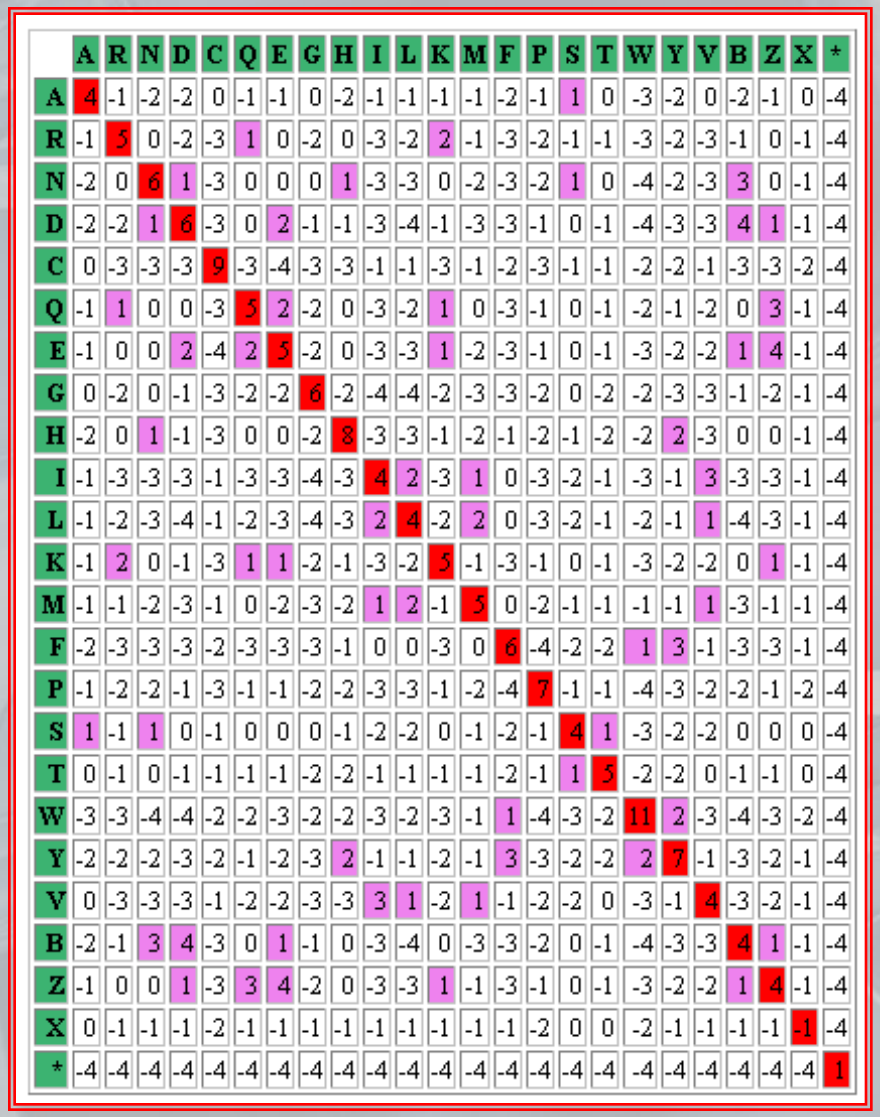
$$B_{ij} = \lfloor k \cdot log(M(A_i, A_j)/C(A_i, A_j)) \rfloor, \; k \text{ costant}$$

  - $M(A_i, A_j)$ is the substitution frequency of the amino acid A_j with the amino acid A_i, observed in the block of the considered homologous proteins
  - $C(A_i, A_j)$ is the expected substitution frequency, represented by the product of the substitution frequencies of amino acids A_i and A_j in the totality of the considered blocks of homologous proteins
- Even in this case, the matrix element (i,j) is proportional to the substitution frequency of the amino acid A_j with the amino acid A_i

phylogenetically distant sequences — **phylogenetically close sequences**

BLOSUM35 — **BLOSUM62**

Example: BLOSUM62



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

PAM or BLOSUM – 1

- The two types of matrices start from different assumptions
  - For PAM matrices, it is assumed that the observed amino acid substitutions for large evolutionary distances derive solely by the summing of many independent mutations; the resulting scores express how likely it is that the alignment of a particular pair of amino acids is due to homology rather than to randomness
  - The BLOSUM matrices are not explicitly based on an evolutionary model of mutations; each block is obtained from the direct observation of a family of related proteins (so probably also evolutionarily related), which show a given degree of similarity

PAM or BLOSUM – 2

- An increasing PAM index describes a suitable score for "distant" proteins, expressing also an evolutionary distance; instead, an increasing BLOSUM index represents a suitable score for protein similar to each others, expressing the minimum conservation value for the BLOCK
- PAM matrices tend to reward amino acid substitutions resulting from single base mutations, also penalizing substitutions involving more complex changes in the codons; instead, they do not reward the conservation of structural amino acid motifs, as the BLOSUMs do

PAM or BLOSUM – 3

- The correlation between PAM and BLOSUM, to a comparable level of substitutions, indicates that the two types of matrices produce similar results

  Equivalent PAM and BLOSUM matrices

  PAM1 → Blosum100
  PAM100 → Blosum90
  PAM120 → Blosum80
  PAM160 → Blosum62
  PAM200 → Blosum52
  PAM250 → Blosum45

PAM or BLOSUM – 4

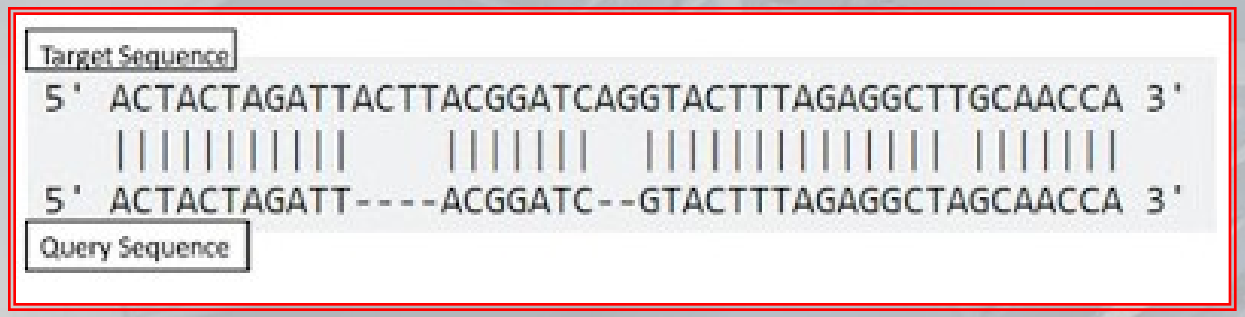
- Typically, the BLOSUM matrices are deemed most suitable to search for sequence similarity
- The BLOSUM62 matrix is normally set as the default in the similarity search software (like BLAST)
- In any case, it is important to choose the most suitable matrix based on the phylogenetic distance between the sequences to be compared
- For phylogenetically close sequences (and organisms) low index PAM or high index BLOSUM must be chosen
- For phylogenetically distant sequences, high index PAM or low index BLOSUM are suitable

Dynamic Programming: The Needleman-Wunsch algorithm – 1

- Once having selected a method for assigning a score to an alignment, it is necessary to define an algorithm to determine the best alignment(s) between two sequences
- The exhaustive search among all possible alignments is generally impractical
  - For two sequences, respectively 100 and 95 nucleotides long, there are ~55 millions possible alignments, just only in the case of five gaps inserted in the shorter sequence
- The exhaustive search approach becomes rapidly intractable
  - Using dynamic programming, the problem can be divided into subproblems of more "reasonable" size, whose solutions must be recombined to form the solution of the original problem

Dynamic Programming: The Needleman-Wunsch algorithm – 2

- S. B. Needleman and C. D. Wunsch, in 1970, were the first who solved this problem with an algorithm able to find global similarities, in a time proportional to the product of the lengths of the two sequences
- The key for understanding this approach is to observe how the alignment problem can be divided into subproblems



Dynamic Programming: The Needleman-Wunsch algorithm – 3

- Example (to be continued)
  Align **CACGA** and **CGA** with the assumption of uniformly penalizing gaps and mismatches
  - Possible choices to be made with respect to the first character:
    1) Place a gap in the first place of the first sequence (counterintuitive, given that the first sequence is longer)
    2) Place a gap in the first place of the second sequence
    3) Align the first two characters
  - In the first two cases, the alignment score for the first position will be equal to the gap penalty
  - In the third case, the alignment score for the first position will be equal to the match score
  - The rest of the score will depend, in all the cases, on the way in which the remaining part of the two sequences will be aligned

Dynamic Programming: The Needleman-Wunsch algorithm – 4

- Example (to be continued): Align **CACGA** and **CGA**

First position	Score	Sequences to be aligned
C **C**	+1	**ACGA** **GA**
– **C**	–1	**CACGA** **GA**
C **–**	–1	**ACGA** **CGA**

- If we knew the score of the best alignment for the remaining parts of the sequences, we could easily calculate the best overall score relative to the three possible choices

Dynamic Programming: The Needleman-Wunsch algorithm – 5

- Example
  Starting from the assumption of aligning the two initial characters (without inserting gaps), it remains to calculate the alignment score for the sequences **ACGA** and **GA**
- In this operation, it will often be necessary to calculate scores for subsequences
- Dynamic programming is based on constructing a table, in which the partial alignment scores are stored, in order to avoid to recalculate them many times

Dynamic Programming: The Needleman-Wunsch algorithm – 6

- The dynamic programming algorithm computes the optimal alignment between two sequences filling a table with partial scores
  - The horizontal and vertical axes describe, respectively, the two sequences to be aligned
- Example: Table for the alignment of **ACAGTAG** and **ACTCG**, with a gap penalty of –1 and a score of mis/matching equal to 0/1

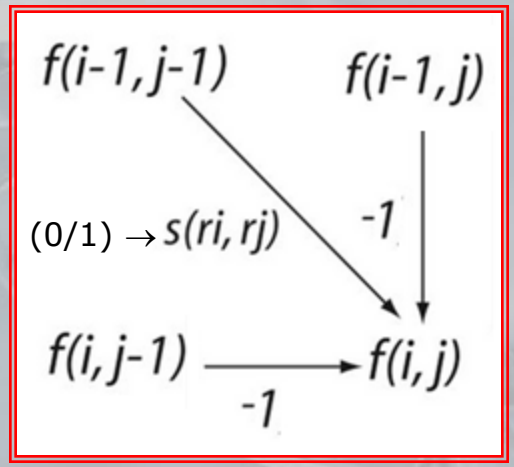
		A	C	T	C	G
	0	–1	–2	–3	–4	–5
A	–1					
C	–2					
A	–3					
G	–4					
T	–5					
A	–6					
G	–7					

Dynamic Programming: The Needleman-Wunsch algorithm – 7

- The alignment of the two sequences is equivalent to build a path that goes from the upper left to the lower right corner of the table
  - A horizontal shift represents a gap inserted in the vertical sequence and vice versa
  - Moving along the diagonal means aligning the corresponding nucleotides in the two sequences
- The first row and the first column of the table are initialized with multiples of the gap penalty (in fact, each gap adds a penalty to the total alignment score)

Dynamic Programming: The Needleman-Wunsch algorithm – 8

- How can we calculate the other elements in the table?
- The element in position (2,2) is calculated by exploring the following three possibilities:
  1) Adding up the gap penalty to the entry in position (2,1), which corresponds to consider a gap in the vertical sequence
  2) Adding up the gap penalty to the entry in position (1,2), which corresponds to consider a gap in the horizontal sequence
  3) Adding up the mis/match score to the entry in the diagonal position (1,1), which corresponds to the alignment of the related nucleotides
- The maximum value among those obtained for the three options (–2,–2,1) is then assigned to the element in position (2,2)



Dynamic Programming: The Needleman-Wunsch algorithm – 9

- We can then proceed to fill the entire second row, then move on to the next row, up to complete the table

		A	C	T	C	G
	0	–1	–2	–3	–4	–5
A	–1	1	0	–1	–2	–3
C	–2	0	2	1	0	–1
A	–3	–1	1	2	1	0
G	–4	–2	0	1	2	2
T	–5	–3	–1	1	1	2
A	–6	–4	–2	0	1	1
G	–7	–5	–3	–1	0	2

- Example

$$N(3,5) = max\{(+1-1),(-2-1),(-1+1)\}$$
$$= max\{0,-3,0)\} = 0$$

Dynamic Programming: The Needleman-Wunsch algorithm – 10

- After completing the table, the value in the lower right corner is the score for the optimal alignment between the two sequences (2, in the example)
- Remark: The score was determined without having to assign a score to all the possible alignments between the two sequences
- The table of the partial scores allows to reconstruct the optimal alignments (generally more than one) between the two sequences
  - Tracing a path from the lower right to the upper left position

Dynamic Programming: The Needleman-Wunsch algorithm – 11

- Example

  The value N(8,6)=2 may have been obtained following three different routes, but the only one that can produce a value of 2 is that coming from N(7,5)=1 (alignment of **G** in both the sequences)

- Again, for the value N(7,5) exists only one possibility, which leads to the element N(6,4)=1 (with a 0 as the mismatch score between the two nucleotides)
- The process must be re-peated until all the possible paths are completed, to reach the final position (1,1)

		A	C	T	C	G
	0	–1	–2	–3	–4	–5
A	–1	1	0	–1	–2	–3
C	–2	0	2	1	0	–1
A	–3	–1	1	2	1	0
G	–4	–2	0	1	2	2
T	–5	–3	–1	1	1	2
A	–6	–4	–2	0	1	1
G	–7	–5	–3	–1	0	2

Dynamic Programming: The Needleman-Wunsch algorithm – 12

- If n and m represent the lengths of the two sequences to be aligned, to convert a path in an alignment, each path from $(n+1,m+1)$ to (1,1) must be traveled backwards, recalling that:
  - a vertical movement represents a gap in the sequence along the horizontal axis
  - a horizontal movement represents a gap in the sequence along the vertical axis
  - a diagonal movement represents an alignment of the nucleotides, belonging to the two sequences, at the current position

Dynamic Programming: The Needleman-Wunsch algorithm – 13

- Example

G
G
CG
AG
TCG
TAG
–TCG
GTAG
––TCG
AGTAG
C––TCG
CAGTAG
AC––TCG
ACAGTAG

		A	C	T	C	G
	0	–1	–2	–3	–4	–5
A	–1	1	0	–1	–2	–3
C	–2	0	2	1	0	–1
A	–3	–1	1	2	1	0
G	–4	–2	0	1	2	2
T	–5	–3	–1	1	1	2
A	–6	–4	–2	0	1	1
G	–7	–5	–3	–1	0	2

Remark: Following all the paths $(n+1,m+1)\rightarrow(1,1)$ in the table of the partial scores, all the possible optimal alignments between the two sequences can be reconstructed

The Needleman-Wunsch algorithm Example – 1

- Alignment of the sequences **CACGA** and **CGA**

		C	G	A
	0	–1	–2	–3
C	–1	1	0	–1
A	–2	0	1	1
C	–3	–1	0	1
G	–4	–2	0	0
A	–5	–3	–1	1

N(2,2) = $max\{(-1-1),(-1-1),(0+1)\} = max\{-2,-2,1)\} = 1$
N(2,3) = $max\{(1-1),(-2-1),(-1+0)\} = max\{0,-3,-1)\} = 0$
N(2,4) = $max\{(0-1),(-3-1),(-2+0)\} = max\{-1,-4,-2)\} = -1$
N(3,2) = $max\{(-2-1),(1-1),(-1+0)\} = max\{-3,0,-1)\} = 0$
N(3,3) = $max\{(0-1),(0-1),(1+0)\} = max\{-1,-1,1)\} = 1$
N(3,4) = $max\{(1-1),(-1-1),(0+1)\} = max\{0,-2,1)\} = 1$
N(4,2) = $max\{(-3-1),(0-1),(-2+1)\} = max\{-4,-1,-1)\} = -1$
N(4,3) = $max\{(-1-1),(1-1),(0+0)\} = max\{-2,0,0)\} = 0$
N(4,4) = $max\{(0-1),(1-1),(1+0)\} = max\{-1,0,1)\} = 1$
N(5,2) = $max\{(-4-1),(-1-1),(-3+0)\} = max\{-5,-2,-3)\} = -2$
N(5,3) = $max\{(-2-1),(0-1),(-1+1)\} = max\{-3,-1,0)\} = 0$
N(5,4) = $max\{(0-1),(1-1),(0+0)\} = max\{-1,0,0)\} = 0$
N(6,2) = $max\{(-5-1),(-2-1),(-4+0)\} = max\{-6,-3,-4)\} = -3$
N(6,3) = $max\{(-3-1),(0-1),(-2+0)\} = max\{-4,-1,-2)\} = -1$
N(6,4) = $max\{(-1-1),(0-1),(0+1)\} = max\{-2,-1,1)\} = 1$

The Needleman-Wunsch algorithm Example – 2

- Alignment of the sequences **CACGA** and **CGA**

		C	G	A
	0	–1	–2	–3
C	–1	1	0	–1
A	–2	0	1	1
C	–3	–1	0	1
G	–4	–2	0	0
A	–5	–3	–1	1

⇨ Two admissible paths

- Two optimal alignments with score equal to 1

C––GA **––CGA**
CACGA **CACGA**

Global and local alignments – 1

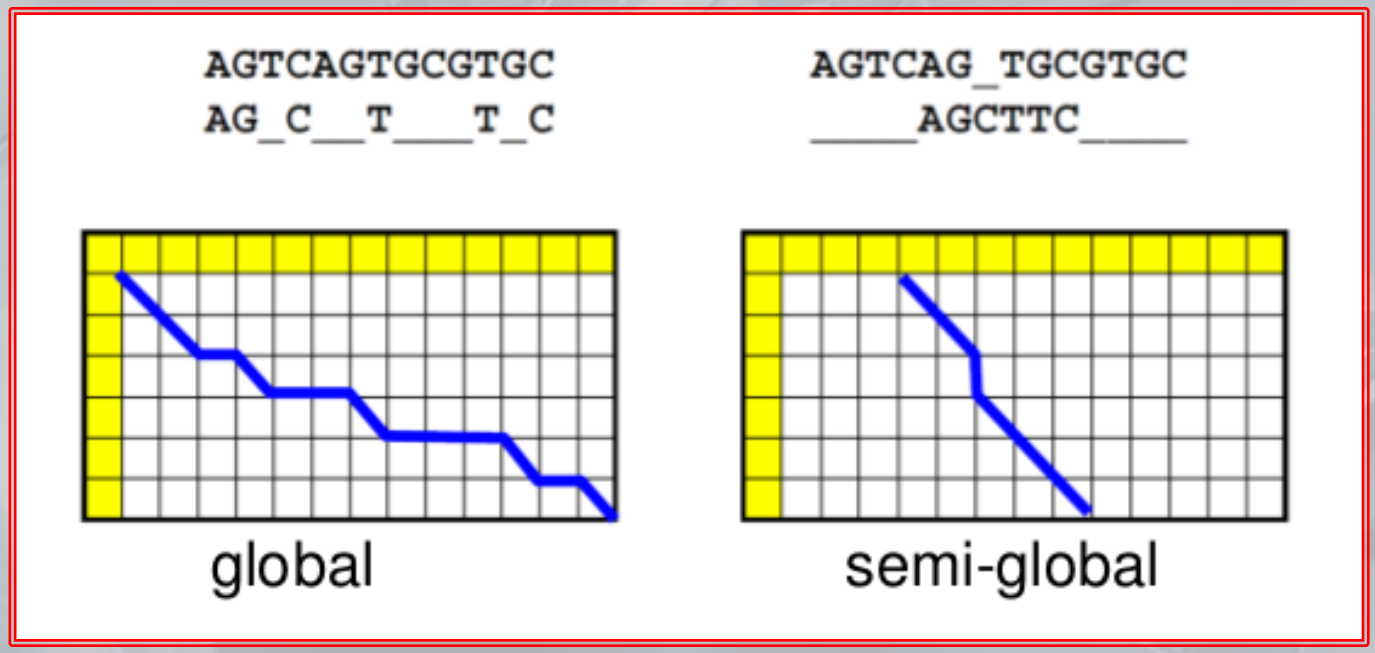
- Global alignment: obtained by trying to align the maximum number of characters between two sequences; ideal candidates are sequences of similar length
- Local alignment: obtained by trying to align "pieces" of sequences with a high degree of similarity; the alignment terminates when "the island of pairing" ends; ideal candidates are sequences with (possibly) significantly different lengths, which contain highly conserved regions

Global and local alignments – 2

- The Needleman-Wunsch algorithm performs global alignments, i.e., it compares sequences in their entirety
  - The gap penalty is fixed, without weighing the gap position (located inside or at the ends of the sequences)
- It is not always the best way to perform the alignment
- Example: let us suppose to search for an occurrence of the short subsequence **ACGT** within the longer sequence **AAACACGTGTCT** (this is a pattern matching approach)
  - Among several possibilities, the alignment of interest is:
    AAACACGTGTCT
    ----ACGT----
  - When searching for the best alignment between a short sequence and a whole genome (to isolate a gene, for instance), penalizing the gaps that appear at one or both the ends of a sequence should be avoided

Global and local alignments – 3

- The flanking gaps are usually the result of an incomplete data acquisition and have no biological significance
  - it is appropriate to treat them differently from internal gaps
  - semiglobal alignment



Global and local alignments – 4

- How can we change the dynamic programming algorithm to wire this new behavior?
- Let us consider again the two sequences **ACTCG** and **ACAGTAG**: we can first move vertically towards the bottom row of the table, and then horizontally to the last column, until we reach the last entry, obtaining:

```
————————ACTCG
ACAGTAG—————
```

- Indeed, from the upper left of the table, each downward movement adds an additional gap at the beginning of the first sequence...
- ...and, since each gap adds a gap penalty to the total score of the alignment, the first column is initialized with the gap penalty multiples

Global and local alignments – 5

- Conversely, if we want to allow the presence of initial gaps in the first sequence without assigning any penalty
  - The first column entries should be set to zero
- Likewise, initializing the first row of the table with all zeroes, we allow the presence of initial gaps in the second sequence without assigning penalties
- Moreover, to admit no gap penalties at the end of a sequence, the meaning of some movement within the table must be differently reinterpreted

Global and local alignments – 6

- Example: Let us suppose to have the following alignment:

  ACACTGATCG

  ACACTG————

  - Using Needleman-Wunsch algorithm to build a path in the table of partial scores, after aligning the first six nucleotides, we reach the bottom row
  - Then, to reach the lower right corner, we should perform four horizontal movements
  - Allow horizontal movements in the last row to have no gap penalties
  - Similarly, vertical movements on the last column should not be penalized

Global and local alignments – 7

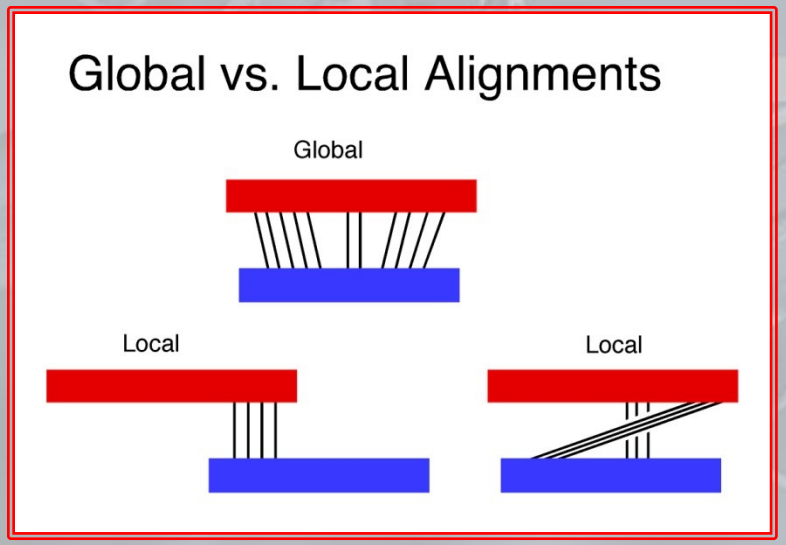
		A	C	A	C	T	G	A	T	C	G
	0	0	0	0	0	0	0	0	0	0	0
A	0	1	0	1	0	0	0	1	0	0	0
C	0	0	2	1	2	1	0	0	1	1	0
A	0	1	1	3	2	2	1	1	0	1	1
C	0	0	2	2	4	3	2	1	1	1	1
T	0	0	1	2	3	5	4	3	2	1	1
G	0	0	0	1	2	3	6	6	6	6	6

Global and local alignments – 8

- In summary:
  - By initializing the first row and the first column of the table with all zeroes…
  - …and allowing non–penalized horizontal and vertical movements, respectively, in the last row and in the last column of the table
  - A semiglobal alignment is performed
- Unfortunately, not even semiglobal alignments offer sufficient flexibility to address all the possible issues related to sequence alignments

The Smith-Waterman algorithm – 1

- In 1981, T. F. Smith and M. S. Waterman developed a new algorithm capable of detecting also local similarity
- Example: Let us suppose to have a long DNA sequence and want to isolate each subsequence similar to each part of the yeast genome
  - A semiglobal alignment is not sufficient because it will however penalize each non–correspondence position
  - Even if there were an interesting subsequence, partly coincident with the yeast genome, all non–correspondent nucleotides will contribute to generate an unsatisfactory alignment score
  - Local alignment



The Smith-Waterman algorithm – 2

- Example: Let us consider the two sequences **AACCTATAGCT** and **GCGATATA**
  - By using a semiglobal alignment with a –1 gap penalty and non/correspondence scores equal to –1/1, we will obtain the following alignment:

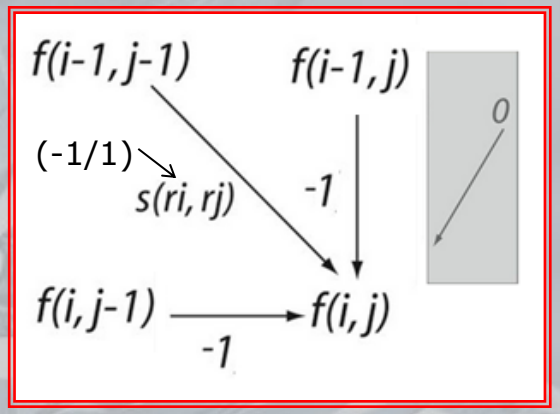
AAC–CTATAGCT
–GCAATATA–––

With the name TATA box, or Goldberg-Hogness box, a canonical sequence (i.e. common to all organisms) is defined, which can be localized on a DNA strand and which forms, together with other canonical sequences such as the CAAT Box or the GC Box, a particular site called the "core" promoter of a gene.

  which is pretty poor, given that four of the top five positions are mismatches or gaps, as well as the last three positions
  - However, there is a "correspondence region" within the two sequences: the **TATA** subsequence
- Change the algorithm in order to identify matches between subsequences, ignoring mismatches and gaps before and after the region(s) of correspondence

The Smith-Waterman algorithm – 3

- For the local alignment of two sequences:
  - Initialize the first row and the first column to zero (as in the semiglobal alignment)
  - Set the mismatch penalty to –1
  - Enter a zero entry in the table wherever all the other routes return a negative score



- After having built the table:
  - Find the maximum partial score
  - Proceed backwards, to rebuild the alignment, until a zero entry is reached
  - The resulting local alignment represents the best matching subsequence between the two given sequences

The Smith-Waterman algorithm – 4

		A	A	C	C	T	A	T	A	G	C	T
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	1	0	0
C	0	0	0	1	1	0	0	0	0	0	2	1
G	0	0	0	0	0	0	0	0	0	1	1	1
A	0	1	1	0	0	0	1	0	1	0	0	0
T	0	0	0	0	0	1	0	2	1	0	0	1
A	0	1	1	0	0	0	2	1	3	2	1	0
T	0	0	0	0	0	1	1	3	2	2	1	2
A	0	1	1	0	0	0	2	2	4	3	2	1

- In summary… when working with long sequences, of several thousands, or millions, of nucleotides, local alignment methods can identify common subsequences, impossible to be found by means of global or semiglobal alignments

A comparison between different alignment algorithms

ATATGGT–
AT–TCGTA

		A	T	A	T	G	G	T
	0	–1	–2	–3	–4	–5	–6	–7
A	–1	1	0	–1	–2	–3	–4	–5
T	–2	0	2	1	0	–1	–2	–3
T	–3	–1	1	0	2	1	0	–1
C	–4	–2	0	–1	1	0	–1	–2
G	–5	–3	–1	–2	0	2	1	0
T	–6	–4	–2	–3	–1	1	0	2
A	–7	–5	–3	–1	–2	0	–1	1

Global alignment
(mismatch score =–2)

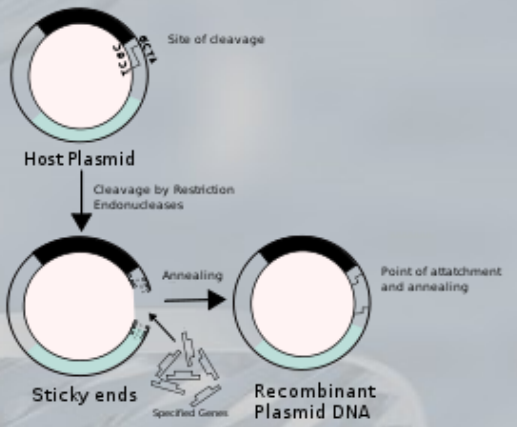
ATAT
AT–T

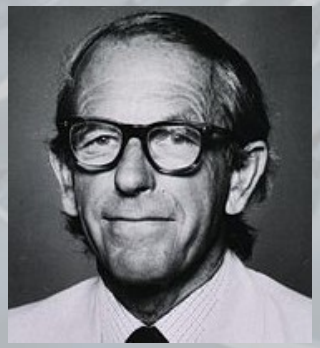
		A	T	A	T	G	G	T
	0	0	0	0	0	0	0	0
A	0	1	0	1	0	0	0	0
T	0	0	2	1	2	1	0	1
T	0	0	1	0	2	1	0	1
C	0	0	0	0	1	0	0	0
G	0	0	0	0	0	2	1	0
T	0	0	1	0	1	1	0	2
A	0	1	0	2	1	0	0	0

Local alignment

Biological data – 1

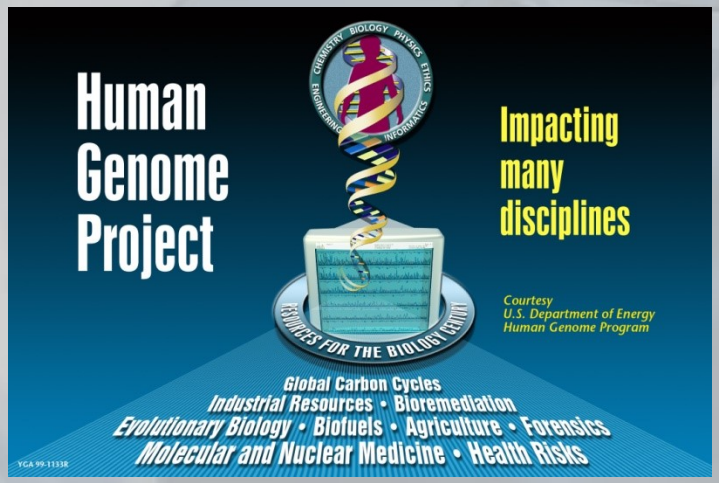
- 1965: Margaret Dayhoff defines an atlas of all known proteins, studying the relationships among their primary sequence; the collected data were distributed in 1970, in the database NBRF (National Biomedical Research Foundation)
- Early '70s: The recombinant DNA technology (based on restriction enzymes and fundamental for cloning) is established, which allows the manipulation of the nucleotide sequences, guaranteeing the comprehension of the DNA structure, function and organization
- Late '70s: Publication of the first genomic data (F. Sanger), with a small number of nucleotide encoding sequences, freely accessible via the network (restricted to few universities)

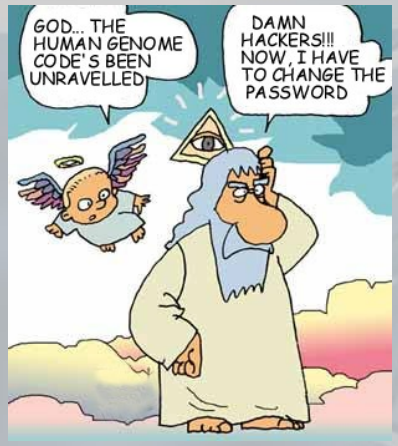




Biological data – 2

- 1980 [Kurt Stueber]: Birth of the first genomic database, at the European Molecular Biology Laboratory (EMBL) in Heidelberg
- 1982 [Walter Goad]: Birth of a similar database in the USA, which will converge later in GenBank
- 1986: A mirror of GenBank, DDBJ (DNA DataBank of Japan), was set up at the National Institute of Genetics in Mishima (Japan)
- 2001: The International Public Consortium and Celera Genomics provide the complete human genome





Biological databases – 1

- Over the past few decades, major advances in the field of molecular biology, coupled with advances in genomic sequencing technologies, have led to an explosive growth in biological information generated by the scientific community
- Biological databases are designed as containers, constructed to store data in an efficient and rational way, and to make them easily accessible to the users
- Their ultimate goal consists in collecting data and providing *ad hoc* tools, available within each database, to analyze them

Biological databases – 2

- Biological databases are libraries of life science information, collected from scientific experiments, high–throughput technologies, computational analyses, and published literature
- Information contained in biological databases includes “raw data”, such as gene and protein sequences, but also protein 3D structures, reports on clinical effects of mutations as well as on similarities among biological sequences and structures, etc.

Biological databases – 3

- Numerous biological databanks exist today:
  - Primary Databanks
    - Nucleotide and amino acid sequences
  - Specialized databanks
    - Genes
    - Protein structures
    - Protein domains and motifs – protein domains are compact semi–independent regions with distinctive functions, linked to the rest of the protein by a portion of the polypeptide chain that serves as a hinge
    - Transcriptome expression profiles – transcriptome (a term analogous to genome, proteome or metabolome) means the set of all transcripts (messenger RNAs) of a given organism or cell type
    - Metabolic pathways – a metabolic pathway (or simply a pathway) is the set of chemical reactions involved in one or more processes of anabolism or catabolism within a cell
    - …

Biological databases – 4

- Therefore, in biological databases, collected information and "raw" data are derived from:
  - laboratory analyses (*in vivo* and *in vitro*)
  - bioinformatic analyses (*in silico*)
  - the literature
- Many biological data are freely downloadable in *flat format*, i.e. in the form of sequential file in which each record is described by one or more consecutive text lines, identified by a particular unique code (a key)
- These files are therefore text files, that can be analyzed by means of suitable tools, able to extract the information of interest
- Alternative: data in HTML or XML format, easy to be consulted via browsers

Biological databases – 5

- Each database is characterized by a central biological element that constitutes the object around which the database records are built
- Therefore, each record collects the information that characterizes the central element (i.e., its attributes)
- A record of a DNA database may contain, in addition to the sequence of a DNA molecule,
  - the name of the organism to which the sequence belongs
  - its functional characteristics (i.e., if it corresponds to a gene or to a non–coding sequence)
  - a list of scientific papers reporting analyses performed on that sequence
  - other interesting information, f.i., in eukaryotes, to which chromosome it belongs

Biological databases – 6

- Biological databases also provide tools for processing the data they contain, including:
  - Query systems (ENTREZ, associated with GenBank, SRS – Sequence Retrieval System, for EMBL, DBGET, for DDBJ)
  - Screening tools (BLAST, FASTA)
  - Multiple sequence alignment tools (ClustalW, Clustal Omega, T–Coffee, ProbCons)
  - Tools for the identification of exons and regulatory elements that characterize a gene (GenScan, Promoser)
  - …

Primary biobanks – 1

- Primary databases contain nucleotide (DNA and RNA) or amino acid (protein) sequences
- The main primary databases are:
  - GenBank (NCBI – *National Center for Biotechnology Information,* founded in 1982 in Bethesda, USA, http://www.ncbi.nlm.nih.gov); the standard database contains 3675462701077 bases belonging to 251998350 sequences (August 2024, genbank/statistics)
  - EMBL datalibrary (founded in 1980 at EMBL – *European Molecular Biology Laboratory*, in Heidelberg, Germany, http://www.embl.de)
  - DDBJ (*DNA DataBase of Japan*, constituted in 1986 by the *National Institute of Genetics* in Mishima, Japan, http://www.ddbj.nig.ac.jp/index-e.html)

Primary biobanks – 2

- Among the three main biological databanks, an international agreement has been established to ensure that DNA data are kept consistent (daily updates made in each bank are automatically transferred to the others)
- Moreover, the three institutions cooperate to share and make publicly available all the data they collect, that differ only in the format in which they are released

NCBI – 1

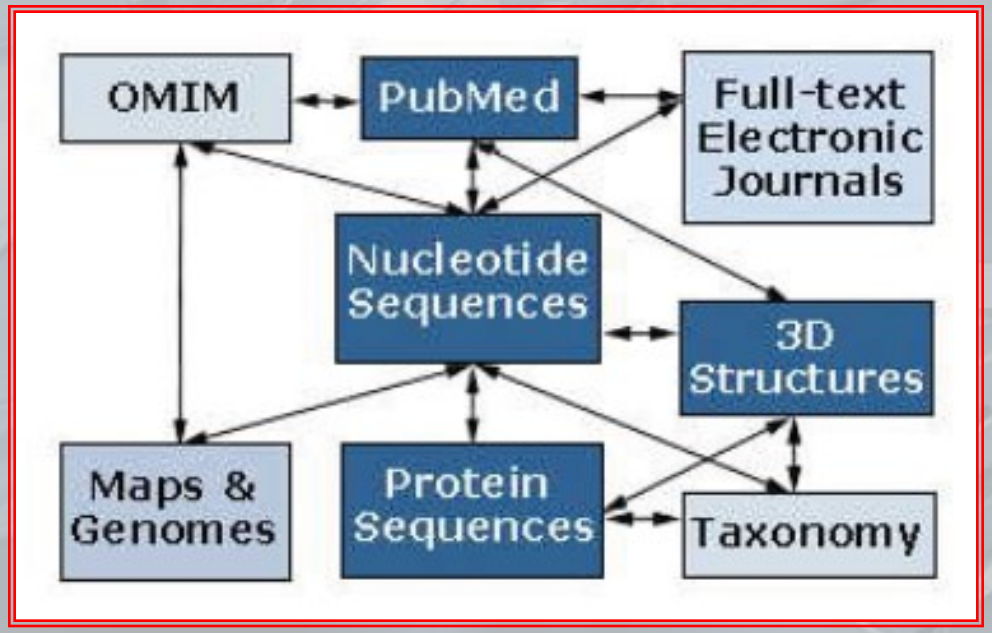
- NCBI is a database of genetic sequences, owned by the USA National Institute of Health; it contains an annotated collection of all publicly available DNA sequences
- Access to data through ENTREZ–Global Query Cross–Database Search System, the query system used for all the different databases managed by NCBI, which therefore constitutes a complete hub to search for information
  - Available via web, for the search and the extraction of information from databases of nucleotide and protein sequences, from the bibliographic database PubMed, the database of Mendelian diseases OMIM, and any database developed by NCBI
  - Closed system: the software that runs the system cannot be downloaded

NCBI – 2

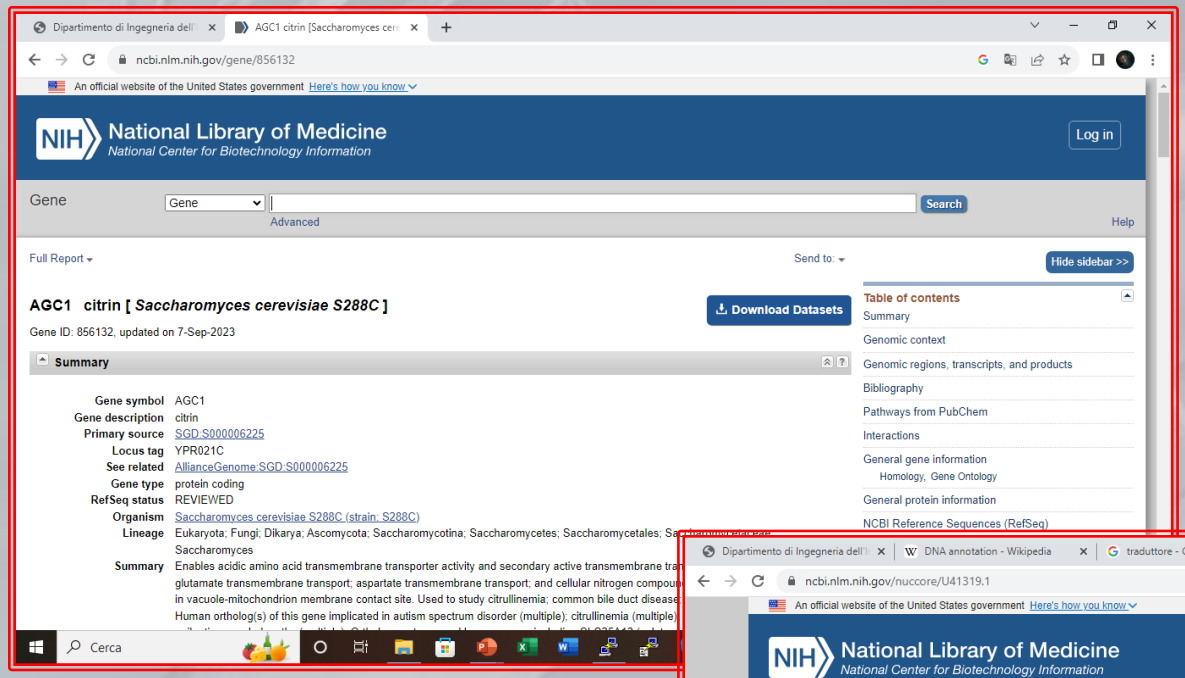
- Main databases in NCBI
  - Nucleotide: It contains the nucleotide non/coding sequences of all the characterized species
  - Gene: It contains data related to the genes of all the characterized species, such as gene structure and genomic context, ontologies, interactions with other genes and links to related sequences and scientific publications
  - Protein: It shares the same structure of Nucleotide, but it contains amino acid sequences
  - PubMed: It is the database of scientific biological and biomedical publications; the abstract is available for each paper; PubMed Central contains full–text articles available for free download

NCBI – 3

- Entrez also provides the possibility to make cross–searching, for collecting information from the various NCBI databases (sequence–structure–genetic map–literature)



NCBI – 4



The AGC1 deficit is a neuro-degenerative syndrome that causes a reduction of the content of myelin, the sheath surrounding nervous cells in the brain. Since the very first months of life, it implies severe psychomotor problems, seizures and difficulties in breathing and in movements controlling.

Protein databanks – 1

- Protein data may be obtained in the following ways:
  - Directly determining the protein sequence
  - Translating the nucleotide sequences for which the function of the encoding gene has been identified or predicted
  - Studying gene expressions
  - Via crystallography, by the determination of secondary and tertiary structures

Protein databanks – 2

- SWISS-PROT (*Protein knowledgebase*, 1986): reference database developed at the Swiss Institute of Bioinformatics (SIB) in Geneve, Switzerland; it contains carefully annotated protein information (often hand–made)
- TrEMBL (*Translated EMBL*): it results from the automatic translation – into amino acid sequences – of all the DNA sequences belonging to the EMBL database and annotated as genes encoding proteins; supplementary to SWISS-PROT
- PIR (*Protein Information Resource)*: mainly devoted to define the annotation standards
- TrEMBL and PIR together (with the European Bioinformatics Institute, 2002) formed the UniProt consortium, the centralized repository of all the protein sequences (http://www.uniprot.org)

Specialized databases

- Specialized databases have been developed later
- They collect sets of homogeneous data from the taxonomic and/or functional point of view, available in primary databases and/or in literature, or derived from experimental approaches, revised and annotated with more information
- Examples:
  - wwPDB (*world wide Protein Data Bank*), the reference database for 3D protein data, equipped with the atomic coordinates determined through X–ray cristallography, NMR analysis, etc.
  - Database of genomic sequences: GDB (man), MGI (mouse), SGD (yeast)
  - Database of Genotypes and Phenotypes (dbGaP) developed to archive and distribute the data and results from studies that have investigated the interaction of genotype and phenotype in humans
  - …

Database search – 1

- Sequence alignments can be a valuable tool for comparing two known sequences
- A more common use of alignments, however, consists in the search within a database, containing biological data, of the sequences that are similar to a particular sequence of interest
- The search results (*target sequences*), which consist of other sequences that align well with (and thus are similar to) the *query sequence*, may in fact provide suggestions on the functional role of the sequence at hand

Database search – 2

- Example: Sequencing of a part of the human genome that could constitute a gene not previously identified
  - comparison of the "putative" gene with millions of sequences deposited in the database Gene at the NCBI
  - clues about its regulation and expression in connection with similar sequences in other species
- During searching in a biological database, the size of both the database and individual data often precludes the obvious approach to align the query sequence to all other sequences, in order to obtain the highest alignment scores
  - Special indexing and search techniques, guided by heuristics, are normally employed

Database search – 3

- Most of the commonly used algorithms do not guarantee to obtain the best alignments, but they do provide some statistical confidence on the retrieval of the majority of the sequences that align well with the query sequence
  - BLAST (*Basic Local Alignment Search Tool*)
  - FASTA (Fast–All, an extension of FAST–N and FAST–P, respectively dedicated to alignments in nucleotide and polypeptide chains)
- The efficiency is a prerequisite and a fundamental feature for these bioinformatic methods, which are of essential support to molecular biologists

BLAST and its variants

- Probably the most popular and commonly used tool to search for sequences in biological databases is BLAST, introduced by S. Altschul *et al.* in 1990
- The original BLAST software looked for long local alignments without gaps, detecting subsequences belonging to the database similar to subsequences of the query sequence
- BLAST can run hundreds of thousands comparisons between sequences in few minutes and, in a short time, a query sequence can be compared with the entire database to search for all the similar sequences
- There are different variants and versions of BLAST, to search for nucleotide and protein sequences
  - BLASTN, BLASTP, BLASTX, TBLASTN (*Translated BLAST Nucleotide*), BLAST+, Magic–BLAST, etc.

BLASTN

- It searches for correspondences among nucleotide sequences, using the simple score matrix:

	A	T	C	G
A	5	–4	–4	–4
T	–4	5	–4	–4
C	–4	–4	5	–4
G	–4	–4	–4	5

with uniform penalties for transitions and transversions, in order to assign scores to alignments without gaps

BLASTP – 1

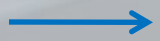
- It searches for correspondence between protein sequences, using PAM or BLOSUM matrices to assign a score to alignments without gaps
  - It divides the query sequence into words, or sub-sequences, of fixed length (4 being the default length)
  - It uses a sliding window, with size equal to the word length, along the entire sequence
    - Example: the query sequence **AILVPTV** produces four different words – **AILV**, **ILVP**, **LVPT**, **VPTV**
  - The words consisting mainly of common amino acids are not considered for searching
  - The remaining words are searched in the database

BLASTP – 2

- When a correspondence is found, the matched sub-sequence is extended in both the directions until the alignment score drops below a given threshold
  - The extension corresponds to the addition of new residues to the matching subsequence with the recalculation of the alignment score in accordance with the scoring matrix
  - The choice of the threshold value is an important parameter because it determines the probability that the resulting sequences are biologically relevant counterparts of the query sequence
  - Example: Search for **AILVPTV**

AILV

MVQGWALYDFLKCRAILVGTVIAML…



AILVPTV

MVQGWALYDFLKCRAILVGTVIAML…

BLAST and its variants (cont.)

- Numerous algorithms for sequence alignment and database search have been developed for specific types of data
  - BLASTN, BLASTX allow, respectively, to search in nucleotide databases and to translate the nucleotide sequence into the protein sequence before searching
  - TBLASTN compares the query protein with the nucleotide sequence database; in order to make this kind of comparison, the database sequences are dynamically translated into amino acid sequences and then compared with the query protein
  - BLAST from 2.0 on (now 2.16.0, June 2024) inserts gaps to optimize the alignment
  - BLAST+ a suite of command–line tools to allow users to run BLAST on their own server without size, volume and database restrictions
  - Magic–BLAST is a tool for mapping large next–generation RNA or DNA sequencing runs against a whole genome or transcriptome

FASTA and its variants – 1

- FASTA algorithms constitute a different family of alignment and search tools
  - They perform local alignments with gaps between sequences of the same type
  - They are more sensitive than BLAST–like algorithms, especially for repetitive query sequences
  - They are computationally more expensive
- Also, in this case the sequence is divided into words
  - of length 4–6 for genomic sequences
  - of length 1–2 for polypeptide sequences
- Successively, a table for the query sequence is constructed, that shows the positions of each word within the sequence

FASTA and its variants – 2

- Example (to be continued)

Let us consider the (query) amino acid sequence

FAMLGFIKYLPGCM

which, for a word of length 1, produces the following table:

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

The column relative to phenylalanine (**F**), for instance, contains the values 1 and 6, which correspond to the positions of **F** in the query sequence

FASTA and its variants – 3

- Example (to be continued)

To compare the query sequence with the target sequence **TGFIKYLPGACT**, a second table is built, with respect to this sequence, that correlates the respective positions of the amino acids

1	2	3	4	5	6	7	8	9	10	11	12
T	**G**	**F**	**I**	**K**	**Y**	**L**	**P**	**G**	**A**	**C**	**T**
	3	–2	3	3	3	–3	3	–4	–8	2	
	10	3				3		3			

Consider the position 2, relative to the first glycine residue (**G**)

- In the query sequence, **G** occupies the positions 5 and 12
- The distances between 5 and 12 and the position of the first **G** in the target sequence (2) produce the two values 3 and 10
- Similarly, in correspondence of the second **G** in position 9, we obtain the values (5–9)=–4 e (12–9)=3

FASTA and its variants – 4

- Example

  Amino acids that are not found in the query sequence, such as threonine (**T**), have not assigned values (the columns in the target table can be deleted)

⇨ The high number of elements with a distance equal to 3 suggests that a shifting of three positions to the left for the query sequence (or of three positions to the right for the target sequence) can produce a reasonable alignment

↓↓↓↓↓↓↓↓

FAMLGFIKYLPGCM

TGFIKYLPGACT

FASTA and its variants – 5

- Comparing the tables, after the *ad hoc* shift of one of the sequences, the identity areas can be found quickly
- These areas constitute an anchor between the query sequence and target sequences found within the database, which are then aligned using the Smith-Waterman algorithm
- However, since the alignment starts from a known region within two similar sequences, FASTA is much faster than the direct use of dynamic programming, which implies to find a complete alignment between the query sequence and all the possible targets

Alignment scores – 1

- Although a database search will always produce a result, without additional information, the extracted sequences cannot always be considered to be related with the query sequence
- The alignment score is the main indicator of how much the search results are similar to the query sequence
  - The alignment scores vary according to the particular search tool
  - Alignment scores do not represent, by themselves, an adequate indicator to establish the actual (evolutionary) correlation between the extracted sequences

Alignment scores – 2

- If the search result gives an alignment score S, we can then ask:
  Given a set of sequences unrelated to the query sequence, which is the probability to randomly find a match with an alignment score equal to S?
- To address this problem, search engines in biological databases provide additional scores, known as E (or E–value) and P, for each output
  - E and P are different because:
    - E is proportional to the expected number of random sequences with an alignment score $\geq S$
    - P represents the probability that the database contains one or more random sequences with score $\geq S$
  - They are closely related and often they have "similar" values

Alignment scores – 3

- Small values for E and P indicate a very low probability that the result of a search has been obtained casually
- Values of $E \leq 10^{-3}$ are considered indicative of statistically significant results
- Often, alignment algorithms provide results with $E \leq 10^{-50}$
  - There is a strong likelihood of evolutionary relationship between the query sequence and the search results

Multiple alignments – 1

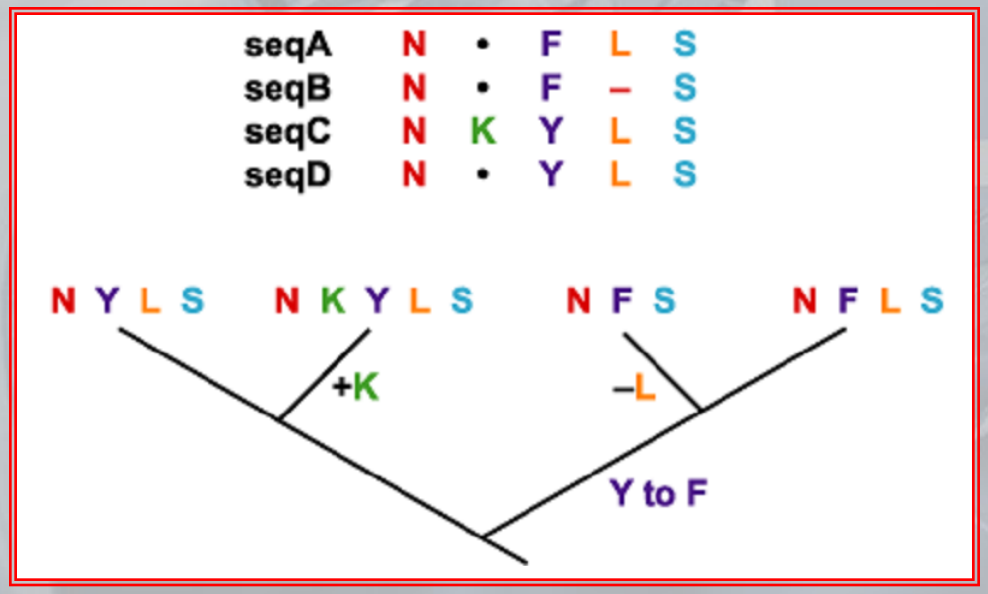
- Multiple alignments are useful when observing a certain number of "similar" sequences, for example to determine the frequencies of substitution
- A multiple alignment can summarize the evolutionary history of a protein family
  - Therefore, we can obtain information about:
    - The conservation of residues dependent on the protein function
    - The conservation of residues dependent on the protein structure
  - Examples of functional/structural information that can be obtained from a multiple alignment:
    - In enzymes, the most conserved regions probably correspond to the active site
    - A conserved pattern of hydrophobic residues alternating with hydrophilic residues suggests a β–sheet
    - A conserved pattern of hydrophobic residues every four residues suggests the existence of an α–helix
- Multiple alignments are also extremely useful for creating score matrices, like PAM and BLOSUM

Multiple alignments – 2

An example of a multiple alignment among sequences

```
1:  EAGFPPGVVNVIPGFGPTAGAAIASHEDVDKVAFTGSTEVGHLIQVA
2:  EAGFPPGVVNIVPGFGPTAGAAIASHEDVDKVAFTGSTEIGRVIQVA
3:  QYMDQNLYLVVKGG-VPETTELL--KERFDHIMYTGSTAVGKIVMAA
4:  NVFSPAWA-TVVEGDETISQQLL--QEKFDHIFFTGSPRVGRLIMAA
5:  EAGVPVGLVNVVQG-GAETGSLLCHHPNVAKVSFTGSVPTGKKVMEM
6:  DI-FPAGVINILFGRGKTVGDPLTGHPKVRMVSLTGSIATGEHIISH
```

Evolutionary significance of a multiple alignment

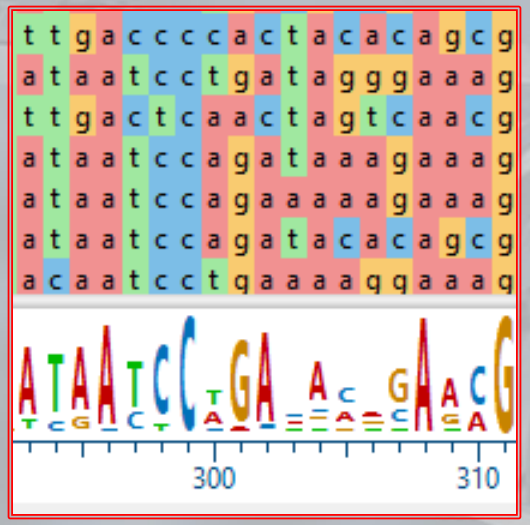


Multiple alignments – 3

- The simplest multiple alignment techniques are logical extensions of the dynamic programming methods (like Needleman-Wunsch)
  - In order to align n sequences an n–dimensional grid is needed
  - The computational complexity of multiple alignment methods grows rapidly with the number of sequences to be aligned
  - Even with a considerable computing power based on massive parallelism, multiple alignments of a few dozen sequences, of medium length and complexity, represent an intractable problem
  - Alignment methods guided by heuristics
    - Clustal

Multiple alignments – 4

- The Clustal algorithm, proposed by Higgins and Sharp in 1988, implements a progressive alignment, trying to match closely related sequences first, and then adding sequences with growing divergence
  - A phylogenetic tree is constructed to determine the degree of similarity among the sequences to be aligned
  - Using the tree as a guide, closely related sequences are aligned in pairs via dynamic programming, to reach the complete multiple alignment



Multiple alignments – 5

- The selection of an *ad hoc* score matrix is fundamental in the case of multiple alignments
  - The use of an inappropriate score matrix will generate a poor alignment
  - Use of a priori knowledge on the similarity degree of the sequences to be aligned
- In ClustalW, the sequences are weighed according to their divergence from the pair of sequences most closely related and the gap penalties and the choice of the score matrix are based on the weight related to each sequence
- Another strategy for multiple alignment is that of not penalizing aligned gaps

Multiple alignments – 6

- Multiple alignments, as well as simple alignments, are based solely on the similarity between nucleotide or amino acid sequences
- The similarity between sequences is an important indicator of functional similarities, even if molecular biologists often have additional knowledge about the structure or the function of a particular gene or protein
  - Information on the secondary structure, on the presence of superficial loops, on the localization of active sites may be used to adjust multiple alignments "by hand", in order to produce biologically significant results

Concluding… – 1

- An alignment of two or more genetic or polypeptide sequences represents a hypothesis on the pathway through which homologous sequences have evolved by diverging from a common ancestor
- While the evolutionary path cannot be deduced with certainty, alignment algorithms can be used to identify "similarities" that have a low probability to occur at random
- The choice of the score function is crucial for the quality of the resulting alignment
  - Use of score matrices, such as PAM and BLOSUM

Concluding… – 2

- The Needleman-Wunsch algorithm, for realizing global alignments, and the Smith-Waterman technique, for local alignments, constitute the fundamental basis on which numerous database search algorithms were built
  - BLAST
  - FASTA
  - Clustal
- These algorithms use indexing techniques, heuristics, and fast comparative methods to get a quick comparison between a query sequence and an entire database