BCH441 – BIOINFORMATICS

SEQUENCE ALIGNMENT

BORIS STEIPE

DEPARTMENT OF BIOCHEMISTRY – DEPARTMENT OF MOLECULAR GENETICS
UNIVERSITY OF TORONTO

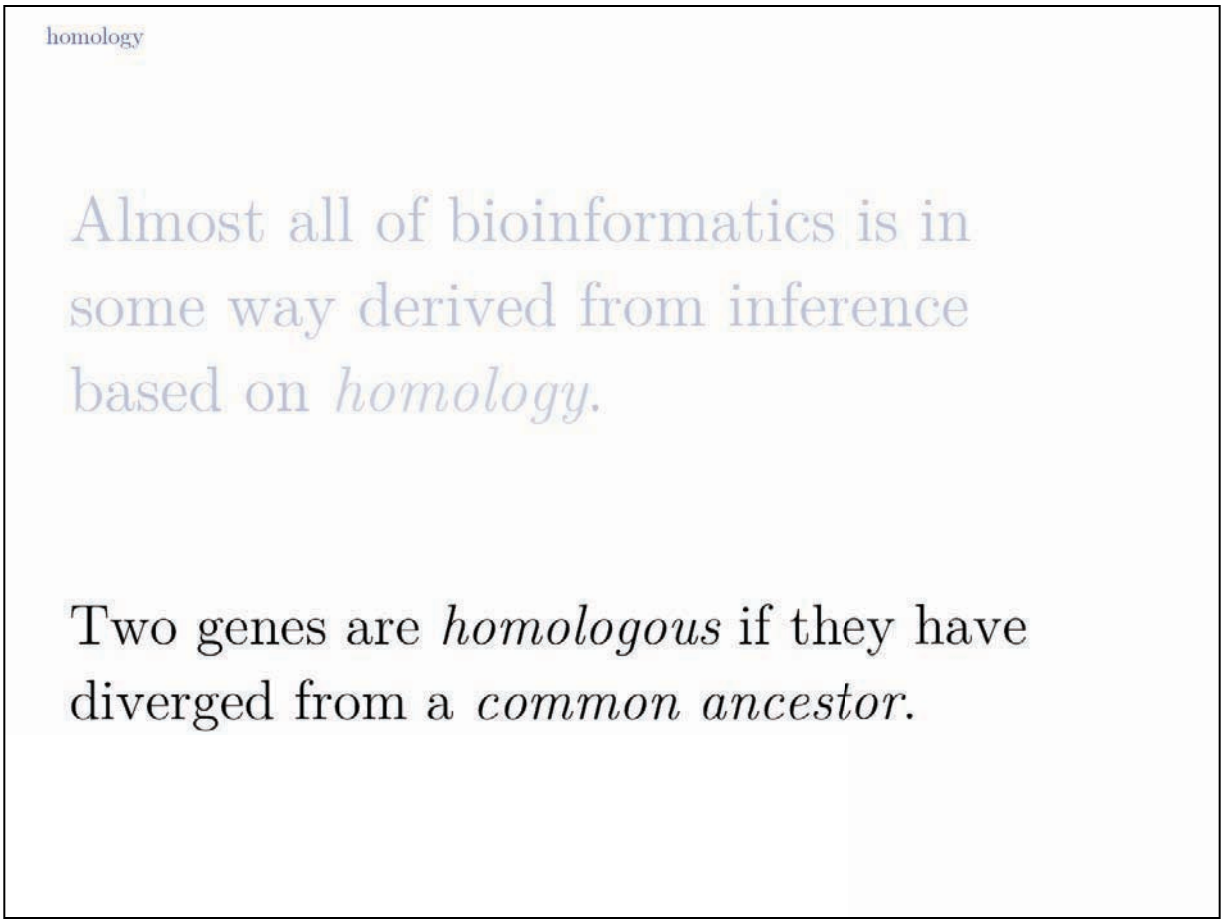
section

Homology

homology

Almost all of bioinformatics is in some way derived from inference based on *homology*.

... and searches for related sequences probably make up the vast bulk of bioinformatics activities.

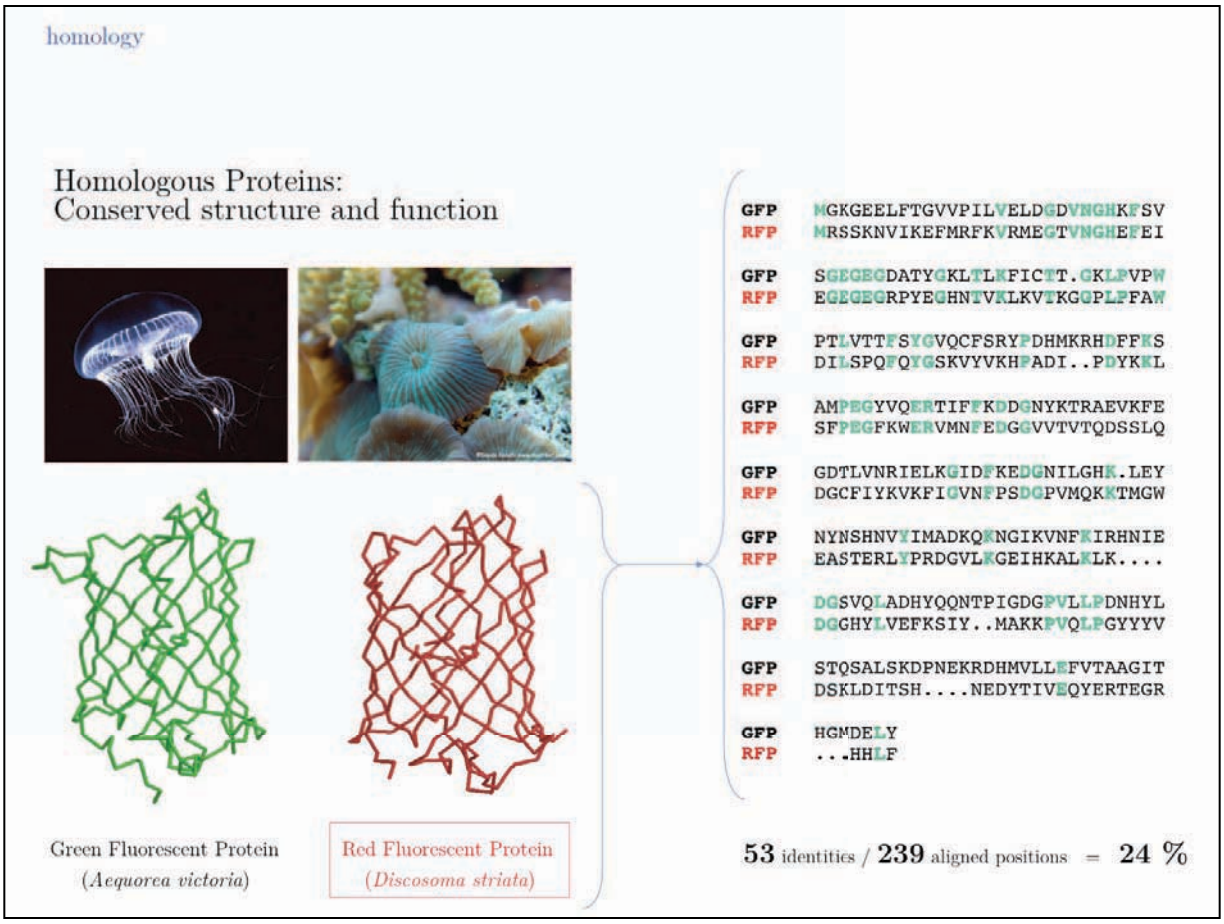


The terms **homology** and **similarity** are often confused and used incorrectly.

Homology is a *quality*. Two genes can either be homologous, or not. There is no such thing is **highly homologous* or 50% homologous. People who speak like that show that they do not fully understand what *homologous* means. Being homologous is like being "pregnant" in that sense: you can only be pregnant, or not – there are no shades in between, like 50% pregnant. This is due to what homology describes: a relationship of **descent from a common ancestor**. Either two genes have a common ancester in their evolutionary history, or they do not. It doesn't make sense to say "common ancestry" over only part of their evolution. However: genes can be – and frequently are – **partially homologous**. This means only a part of their sequence is related, other parts may be related to different genes. As we will discuss later, genes frequently are composed from independently evolving domains.

Similarity on the other hand is a *quantity*. It can be measured, quantified, graded, and compared. Often, **homologous genes have similar sequences**. This implies the possibility to discover homology by measuring sequence similarity.

Also consider the term **analogous**. This is similarty of function or structure or some other property, but not through homology – i.e. descendance from a common ancestor – but by convergent evolution. It is perhaps remarkable that there is no sequence similarity between analogous genes, except for the residues that may be directly involved in a function. (Cf. the analogous versions of hydrolases with a catalytic triad.)



Common ancestry implies similar structure and function.

Many obviously homologous genes have very low similarity. In this example, the aligned sequences of green- and red- fluorescent protein share only 57 of 239 residues, i.e. their pairwise sequence identity is 23.8%. The two organisms share evolutionary ancestry and it is a reasonable hypothesis that the two fluorescent proteins have evolved from the same ancestral sequence. Strikingly, despite 78% amino acid differences in the sequence, the structures of the two proteins are virtually identical and their functions (autocatalytic cyclization and oxidation of a conjugated ssytem of double bonds from a polypeptide precursor) are very similar.

Orthologues:

Genes that have diverged through *speciation*.

Changes on the evolutionary trajectory occur under selective pressure.

Function ususally is conserved.

There are two (and only two) ways to arrive at homologous sequences. Again, there is often confusion about the terms but you really need to know the precise definitions.

Paralogues:

Genes that have diverged through *duplication*.

Changes on the evolutionary trajectory occur under reduced or absent selective pressure.

Consequences:

Function ususally is not conserved:

- *Neo*functionalization
- *Sub*functionalization

*Neo*functionalization: acquisition of a **new** function.

*Sub*functionalization: expression of the **original** function as a response to different signals, during different times, and/or in different tissues.

principles

Homology is not a quantity but a quality.

Homology is commutative.

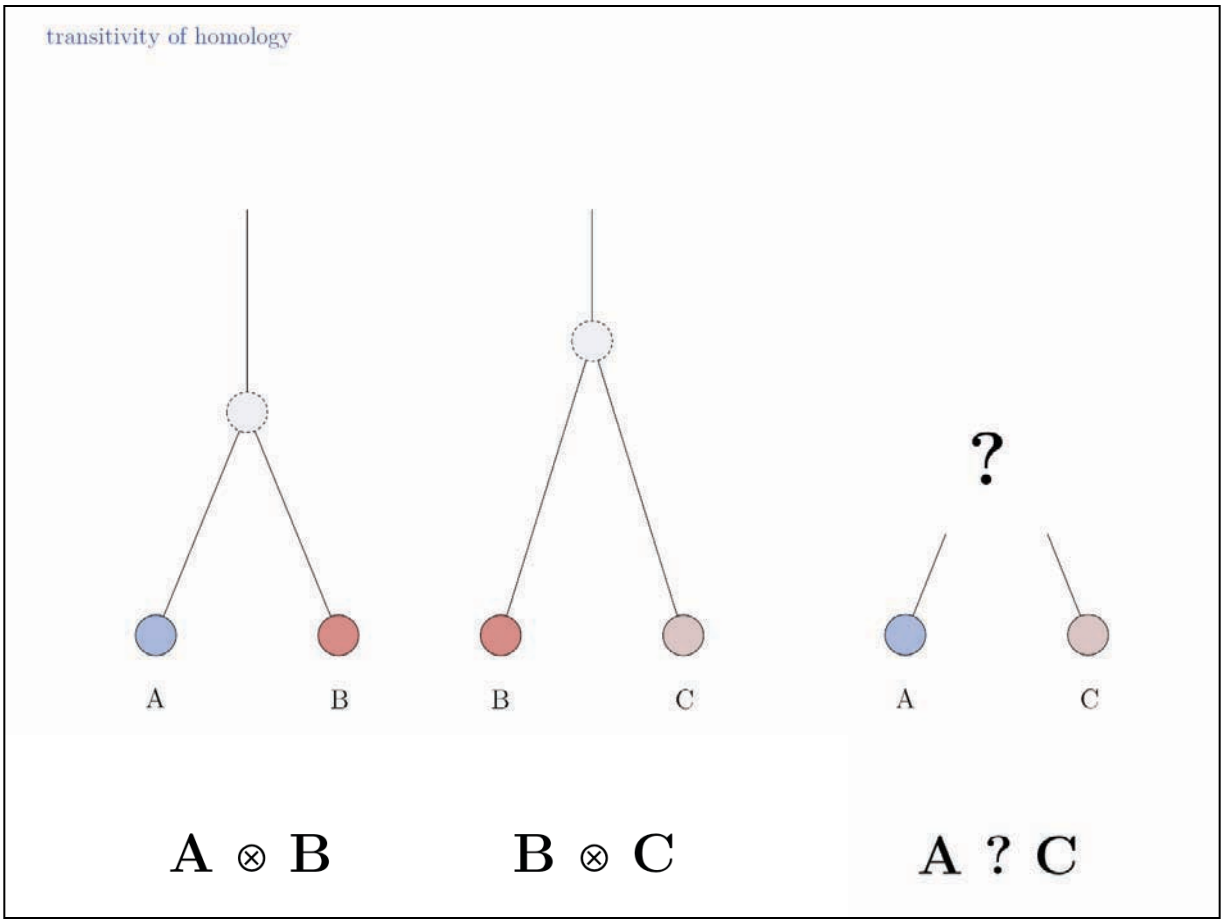
$$A \otimes B \Longrightarrow B \otimes A$$

Homology is transitive.

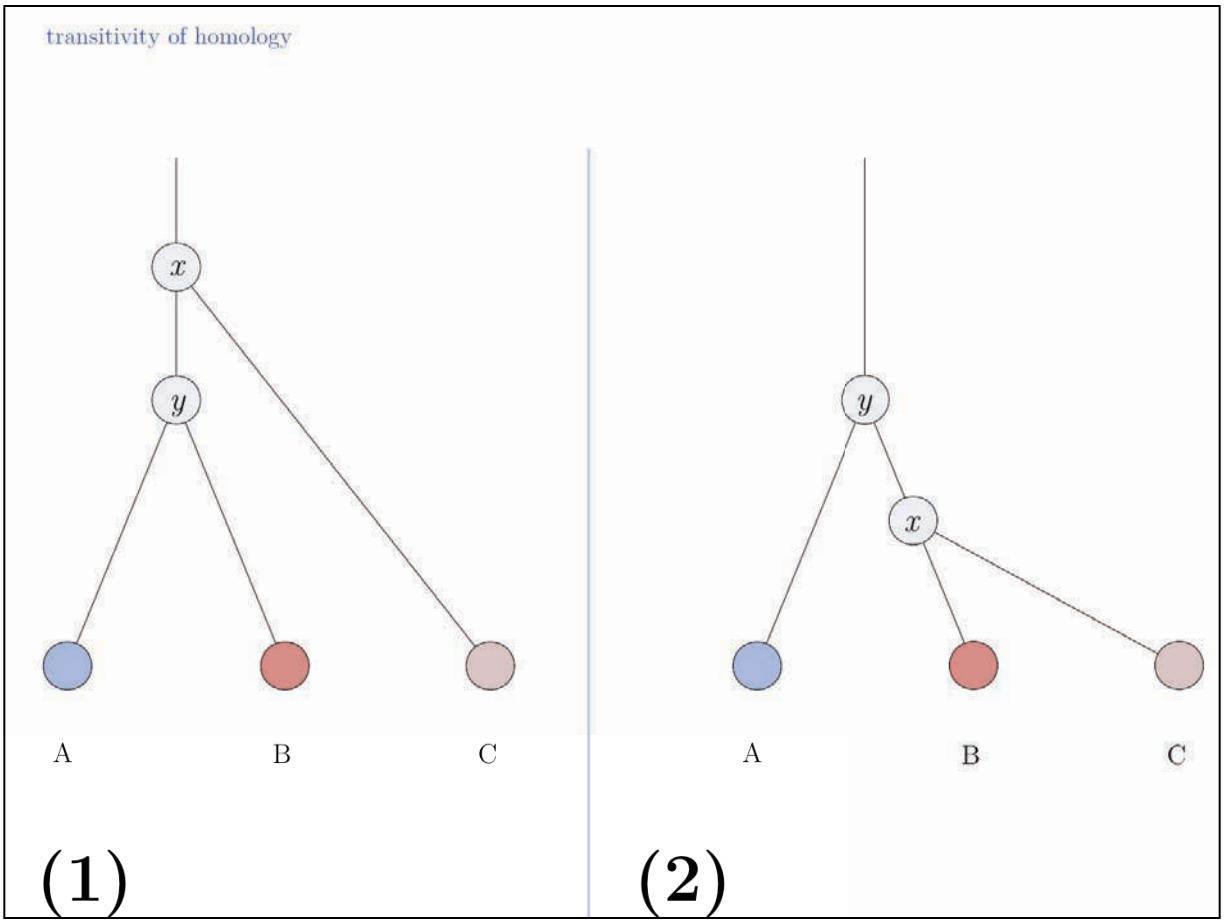
$$A \otimes B, B \otimes C \therefore A \otimes C$$

Three important principles about homology. We have already mentioned the first one, and the second one should be obvious as it is an immediate consequence of the definition.

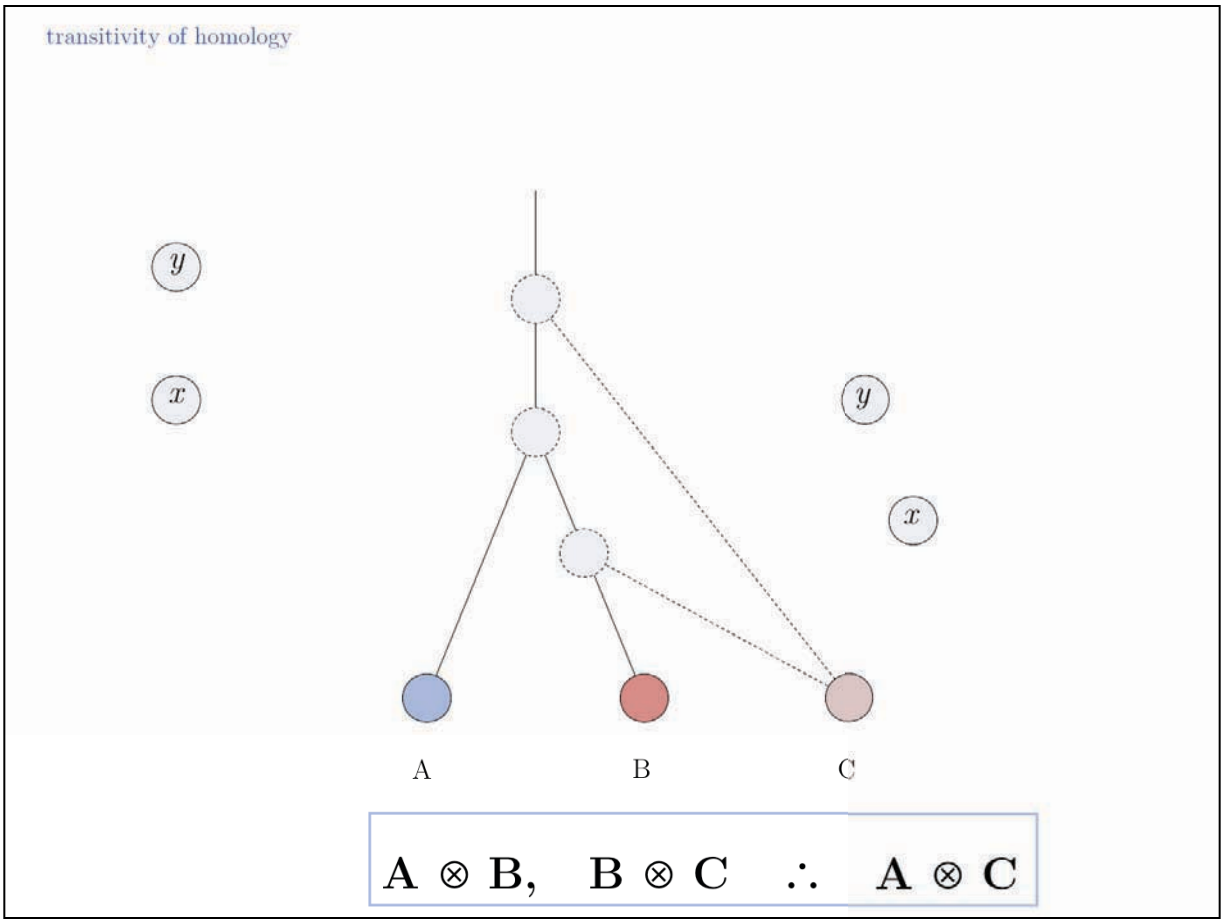
Whether homology also must be transitive requires more a bit more consideration.



Is it necessarily the case that two proteins are homologous if both of them are (perhaps distantly) related to the same third protein?



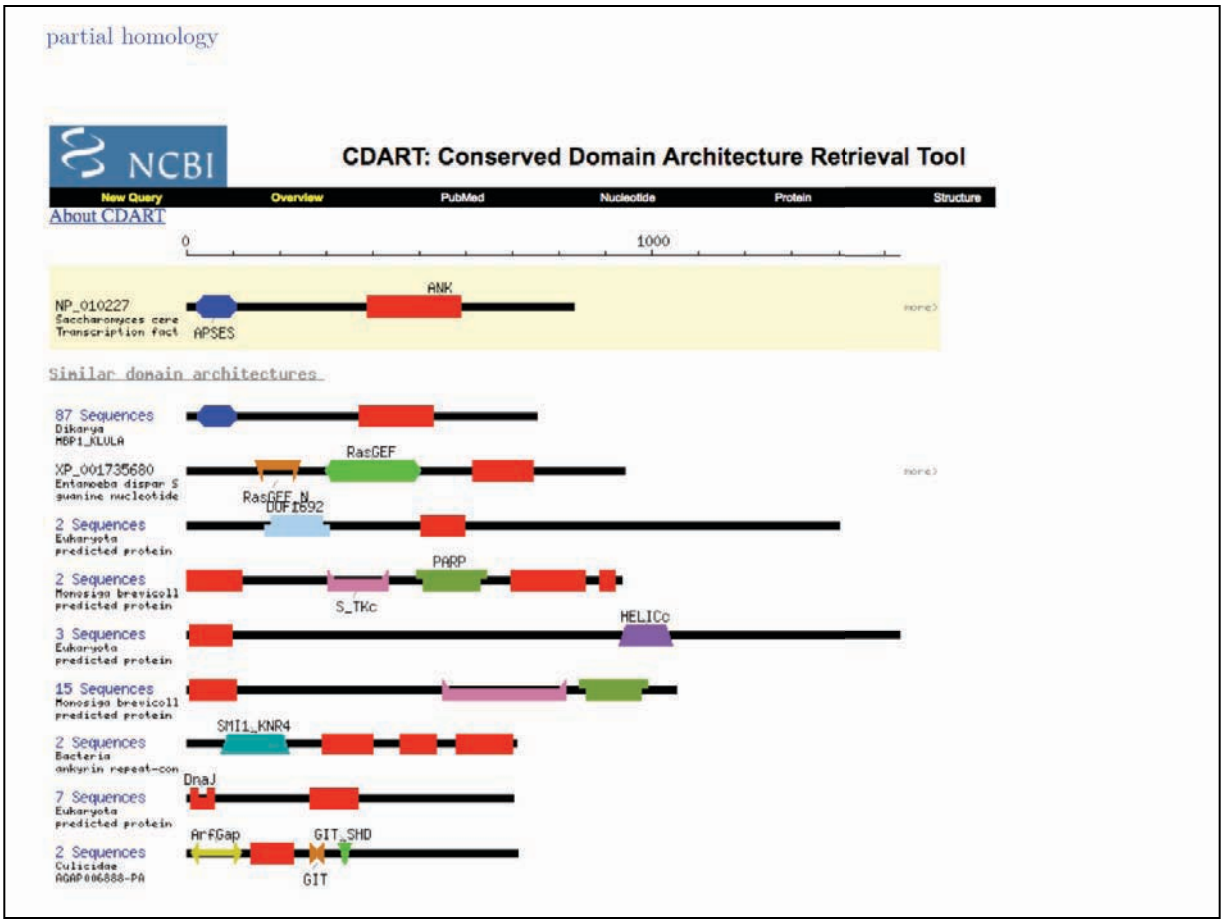
Yes, absolutely. If we draw the evolutionary tree, all three genes are related to the same ancestor. However the ordering of their descent (the topology of their evolutionary tree) may be different: the question is only **where** nodes x and y insert into the tree relative to each other.



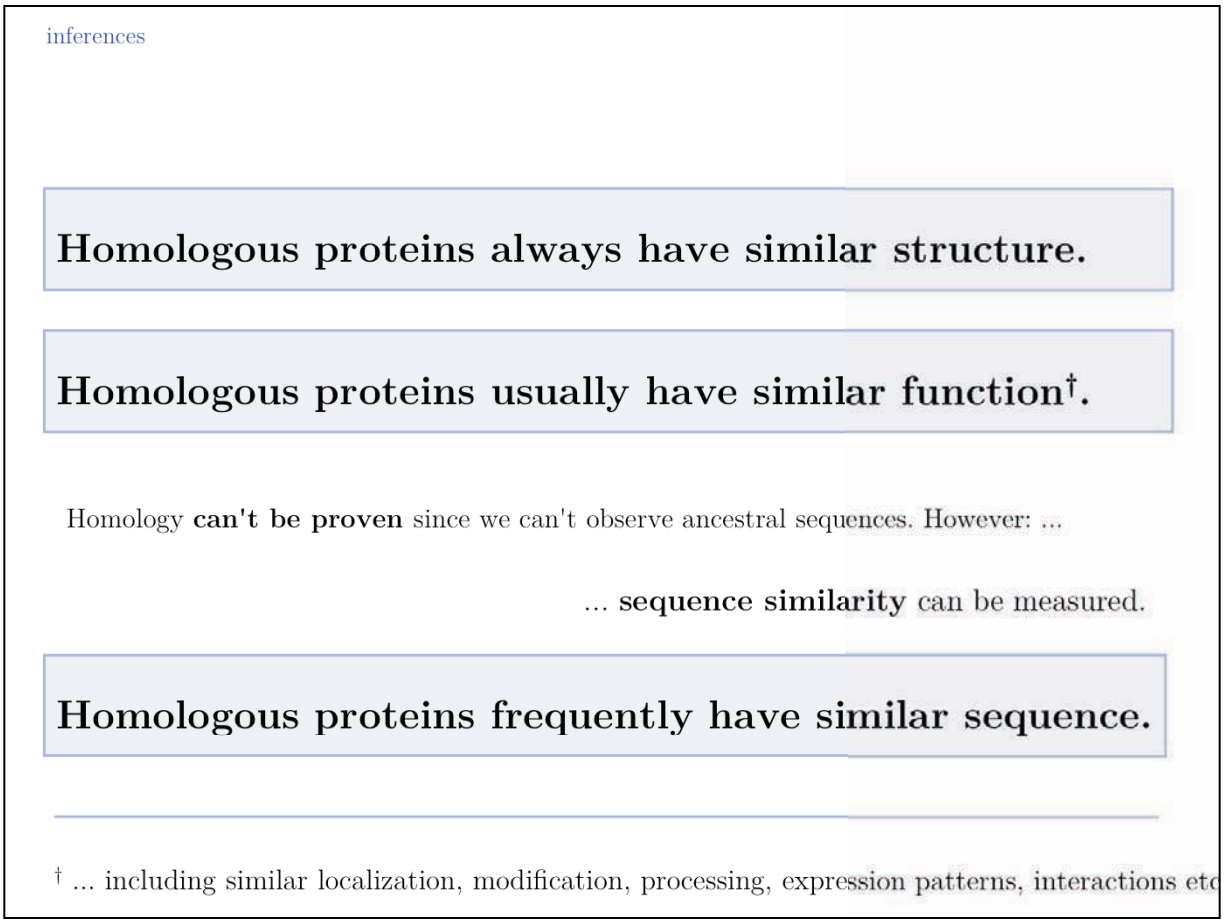
Transitivity of homology justifies inferences across very distant evolutionary relationships, as long as connections via recognizably homologous genes can be defined. This is the basis of advanced alignment algorithms that compare sequences against profiles or probabilistic models: in a group of genes are all homologous if there is a path of homolgy relationships between any pair – however long hat path may be.

But note that this holds **only for domains**, not necessarily for entire genes with their patchwork of (possibly) independently inherited domains.

partial homology



Proteins need not be homologous over their entire length! Each part may have its own, partially independent evolutionary history. Databases such as CDART at the NCBI make this information available.



Homologous proteins always have similar structure.

Homologous proteins usually have similar function†.

Homology can't be proven since we can't observe ancestral sequences.

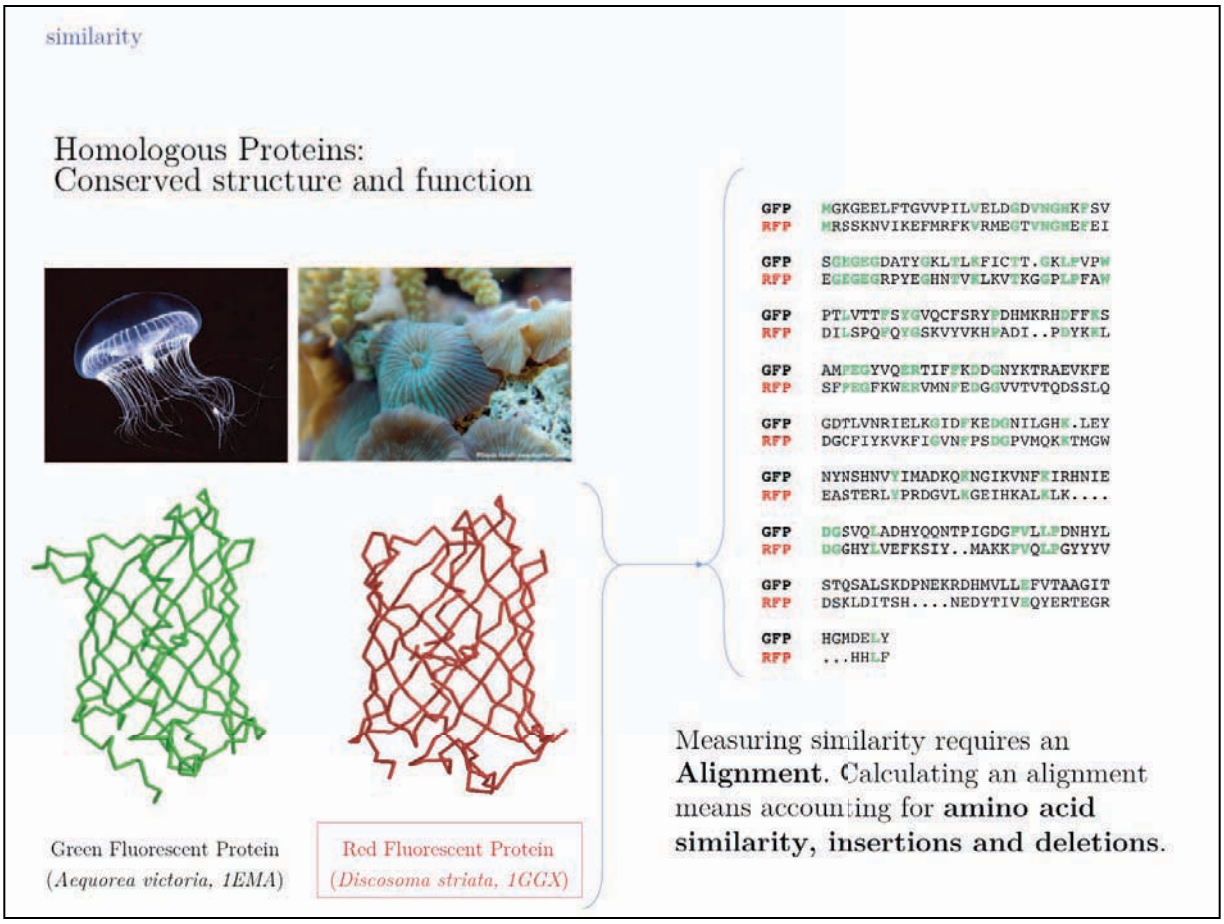
However: sequence similarity can be measured. Homologous proteins frequently have similar sequence.

That said, how do we find sequences that are homologous, or, how do we measure similarity?

† ... including similar localization, modification, processing, expression patterns, interactions etc.

section

Alignment



Inferring homology means measuring similarity.

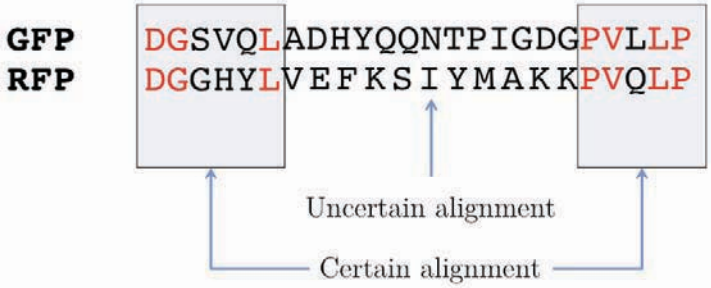
Obviously, the fraction of identical residues depends on the alignment and we need to consider how the right alignment can be obtained. But even before we can start aligning, we need to define a metric for amino acid similarity, because the right alignment should give us good **similarity**, not just a large percentage of **identical** residues. Also, we would like to have a measure that tells us how likely it is that the similarity in an alignment is due to evolutionary descent. And there is an additional issue: how do we treat sequence insertions resp. deletions in the alignment, quantitatively?

alignment

What is an alignment?

What relationship between two amino acids do we want to capture when we write them above each other?

Example: aligning a segment of GFP and RFP with unequal length.



Am alignment is a map of correspondences.

Alignments do not simply consist in writing one sequence above the other and declaring all residues that are in the same column to be related. Proteins evolve to have different lengths through changes at their N- and C- terminus, and internal insertions and deletions (indels). These length changes need to be defined in order to produce an alignment, i.e. the corresponding amino acids are represented by writing one sequence above the other and the correspondence we aim for is to have in each position that pair of amino acids that descended from the common ancestor.

In order for an alignment to make sense, we should strive not to pair-up amino acids that can not be compared on equal terms because they evolve in a very different structural context. But insertions/deletions always change the context over an unpredictable stretch of residues.

Strategies to resolve indels:

```
GFP   DGSVQLADHYQQNTPIGDGPVLLP
RFP   DGGHYLVEFKSIY..MAKKPVQLP
```

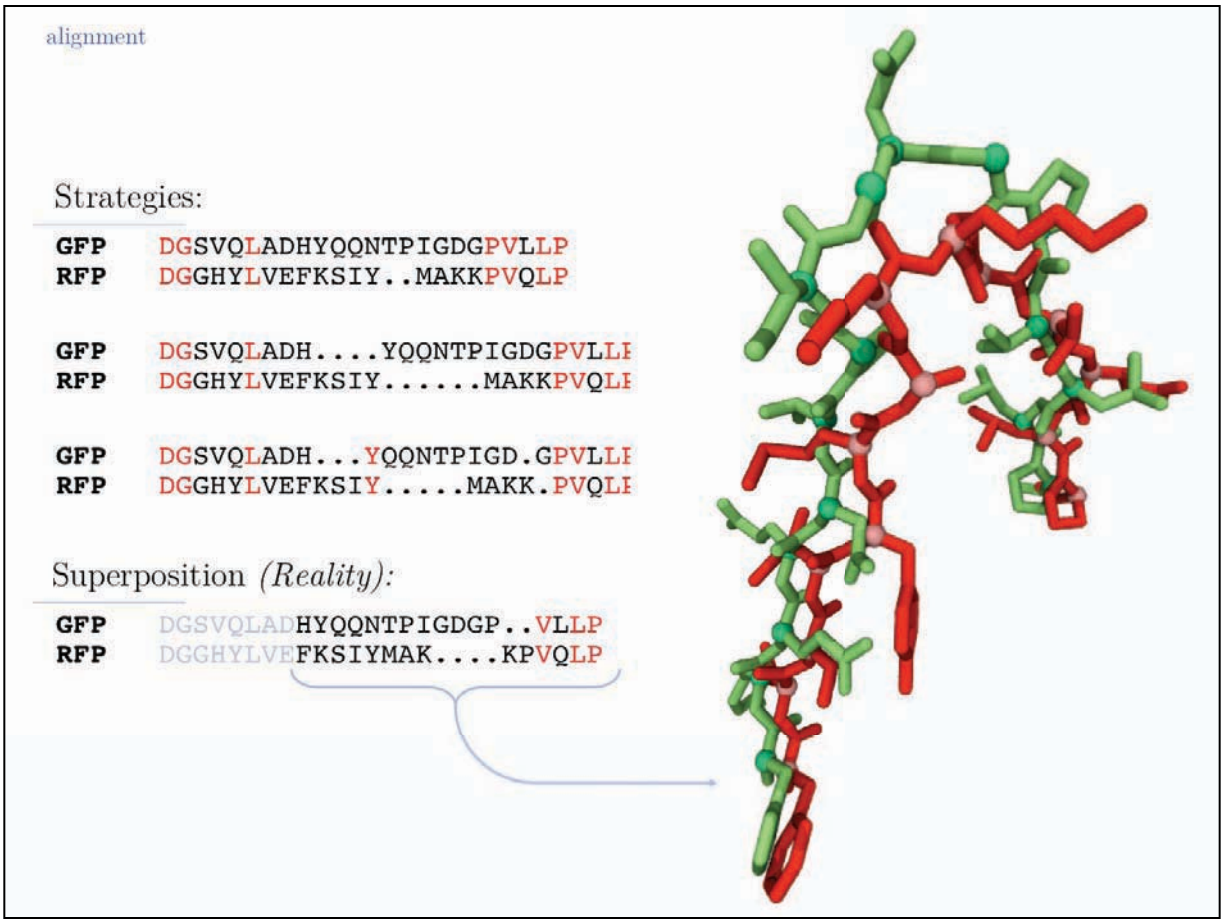
⇦ Minimize gap length

```
GFP   DGSVQLADH....YQQNTPIGDGPVLLP
RFP   DGGHYLVEFKSIY......MAKKPVQLP
```

⇦ Don't align non-equivalent residues

```
GFP   DGSVQLADH...YQQNTPIGD.GPVLLP
RFP   DGGHYLVEFKSIY.....MAKK.PVQLP
```

⇦ Maximize similarity

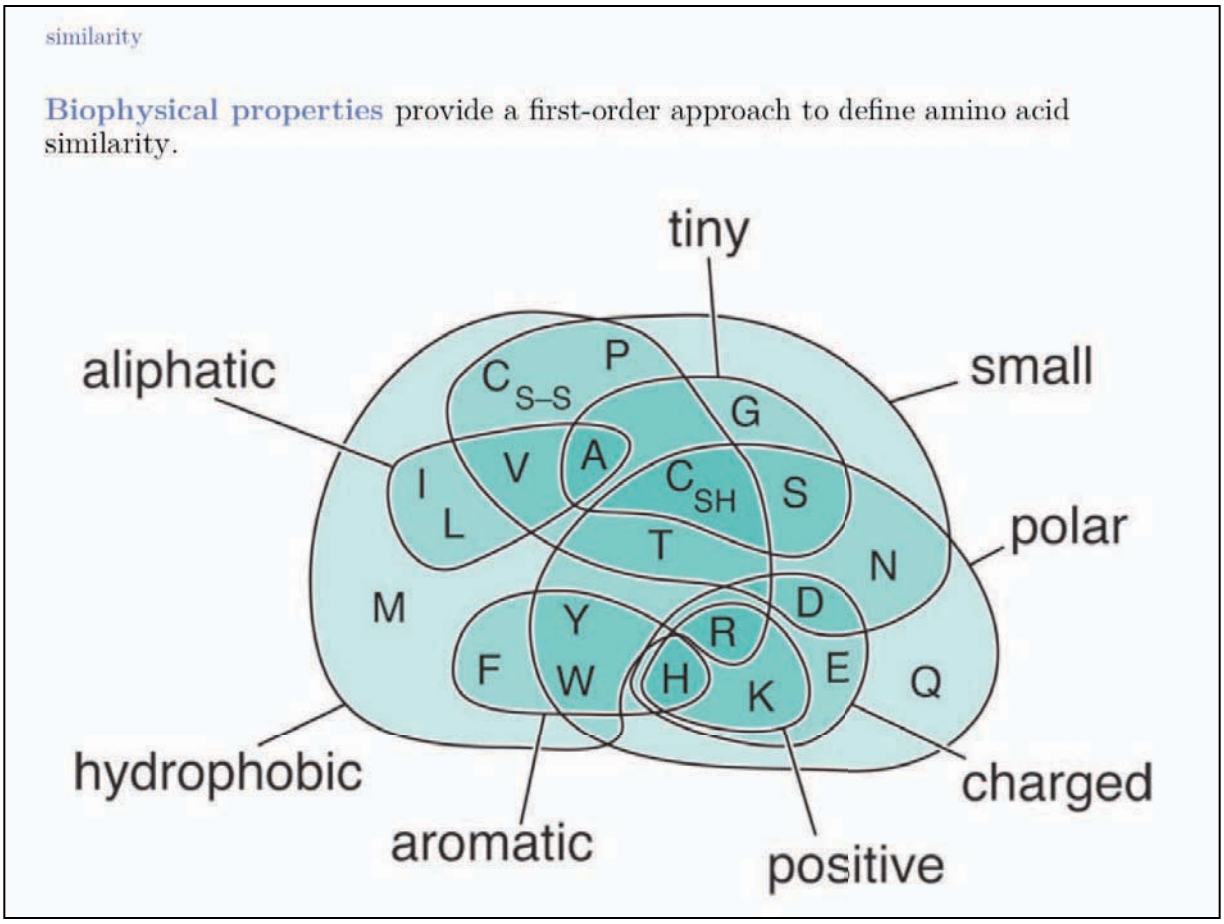


We can consider a structure superposition to be something like the "ground truth" for sequence similarity, it captures the context in which each amino acid performs its function and experiences its selective constraints.

But the superposition does not necessarily capture the **historical process** of a particular sequence change, moreover, it does not necessarily correspond to any of our preconceived heuristic alignments. Part of the problem is that the structural accommodation of an indel is not necessarily the site at which the indel arose during evolution of the sequence.

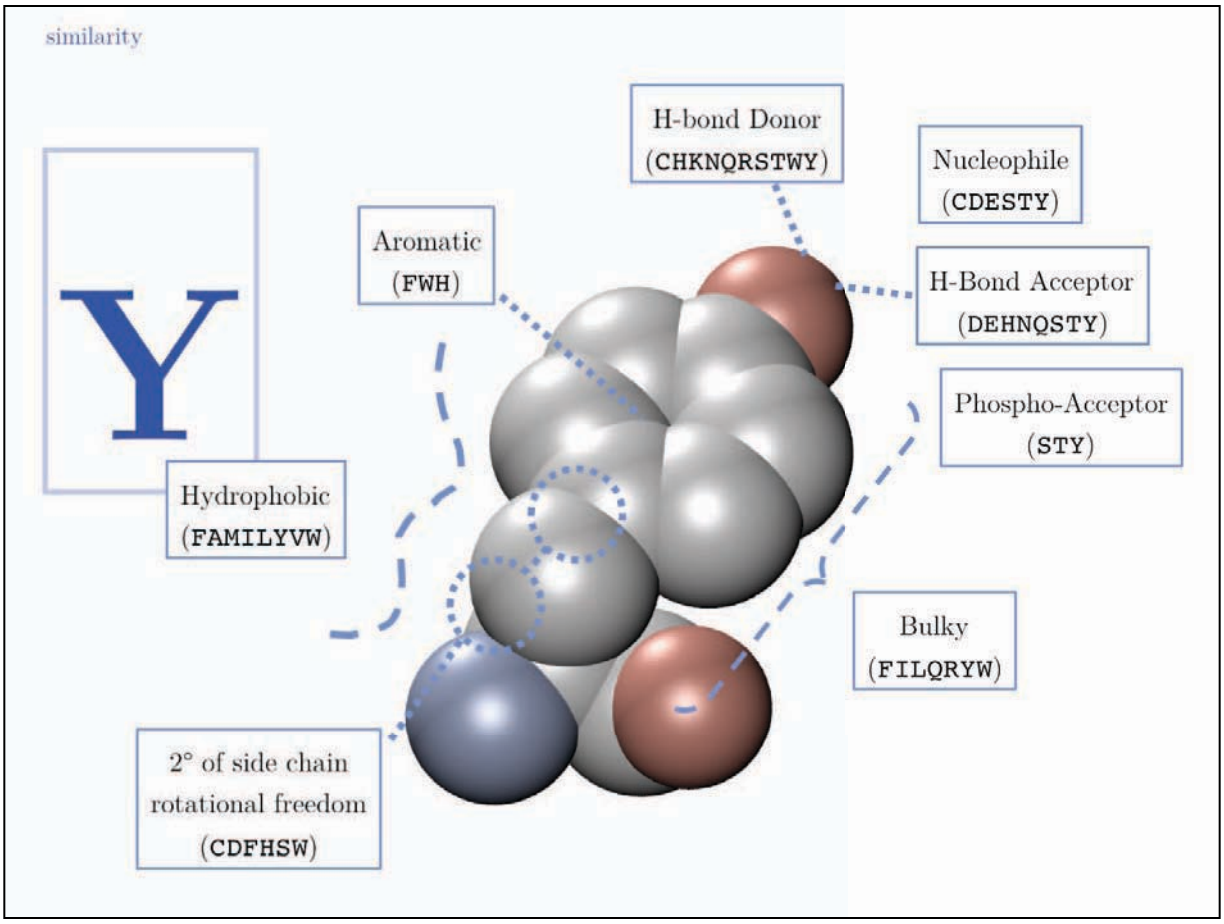
section

Alignment



Obviously, the precise role of a particular amino acid depends on its context in a folded protein, however this Venn diagram (originally going back to Willie Taylor) provides a good first aproximation to summarize shared sidechain properties and to estimate amino acid similarity.

Note that "C" appears twice in this sketch: once as cysteine (C_{SH}) with its free thiol function, once as the disulfide bonded cystine (C_{S-S}). These two forms have very different properties.



Which amino acid(s) we regard as being similar to tyrosine depends on which property we are considering. There are many properties that have been measured, all of them imply "similarity" of different sets of amino acids, and no obvious strategy exists how to define weights to use more than one metric in defining a similarity score for an amino acid pair.

The problem:

Amino acids can have multiple functions.
Which function is important, is determined by **context**.
What is more, context may influence the function.

Quantifying similarity in *sequences* implies a measure based only on pairs of amino acids, independent of the context!

Example:
pK of Side Chain: charge is determined by environment.
E.g. a-helix dipole can easily shift pK by ± 2 pH units ...

pK	AA
3.9	D ASP
4.4	E GLU
6.5	H HIS
9.2	C CYS
10.1	Y TYR
10.5	K LYS
12.0	R ARG

(TJ Creighton, Proteins. 2.ed. Freeman, NY 1993)

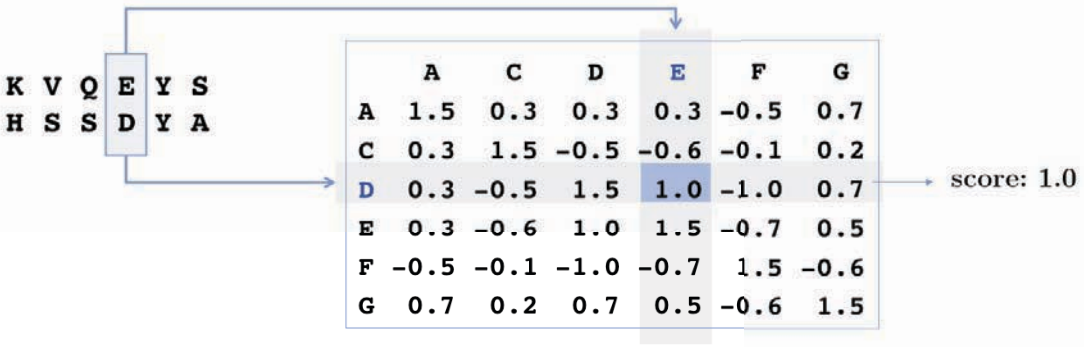
section

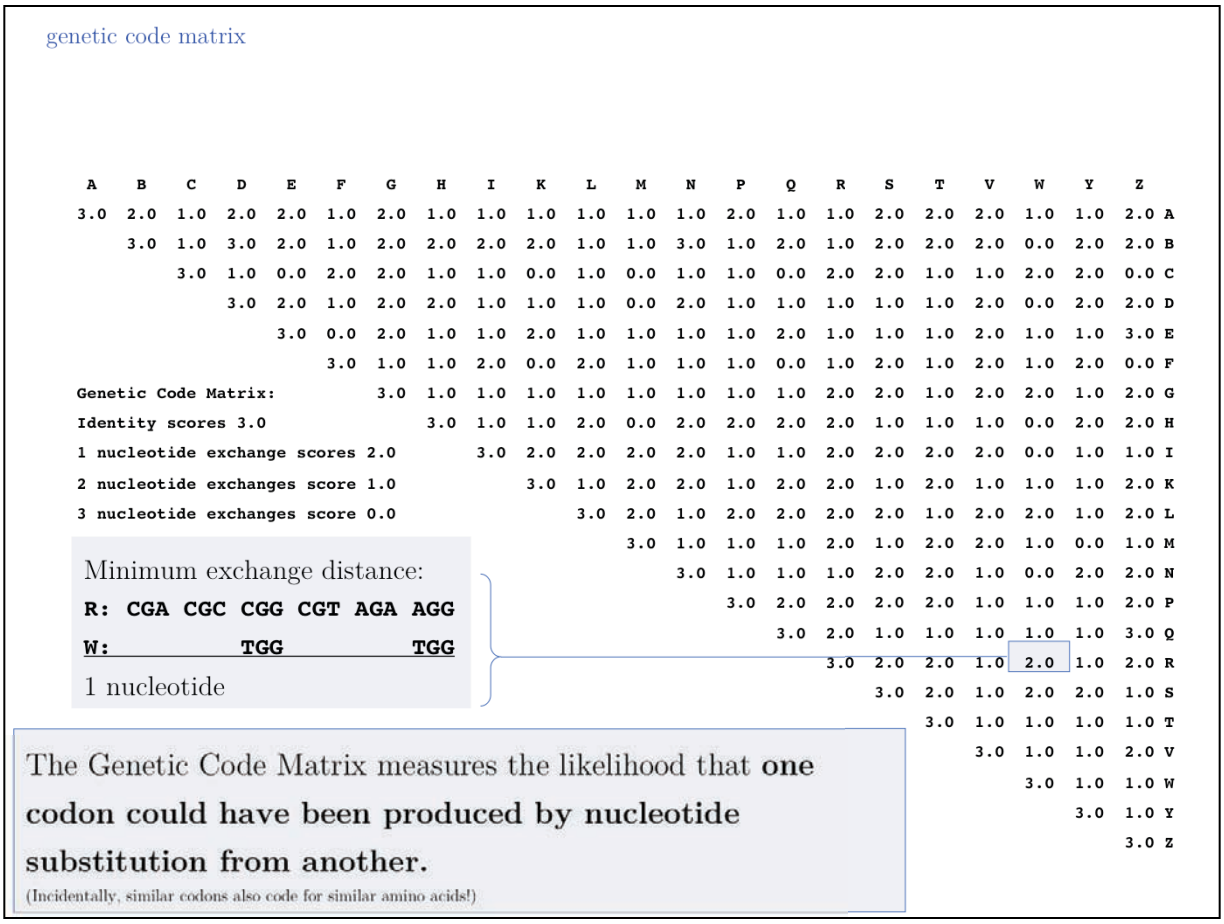
Measuring Similarity

scoring matrix

Models of amino acid similarity can be quantified in a scoring matrix

Scoring matrix:
define similarity of each amino acid with each possible aligned amino acid ...
(also: "similarity matrix", "mutation matrix", "substitution matrix" ...)





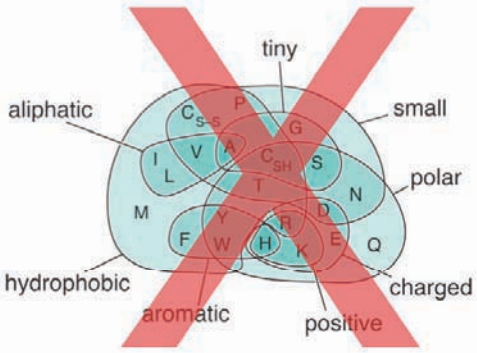
A scoring matrix can be used to quantify how well a given model is represented in two aligned sequences. Here the model says: two amino acids are similar, if it is easy to change one codon into the other by single nucleotide substitutions. For very closely related sequences, this is actually not a bad metric. And it captures an intriguing property of the genetic code: being robust against mutations in the sense that the biophysical properties tend to be conserved between similar codons.

Any biophysical property of amino amino acids can be turned into such a scoring matrix. However, whether amino acids are likely to be paired in a correct alignment of natural sequences is not well described by any single biophysical property, and there is no abvious way how to weight their combinations.

similarity: Dayhoff model

M. Dayhoff

A quantitative model of evolution:



Similarity can be defined as the empirical probability that two amino acids can substitute for each other during evolution!

This makes speculations about about amino acid similarity based on first principles unnecessary.

The Dayhoff model of evolution postulates a quantitative model of the likelihood of specific amino acid substitutions as a consequence of evolution, based on the empirical observation of variation in related protein sequences. This rejects a definition of amino acid similarity from first principles in favor of an empirical approach.

Dayhoff Mutation Data Matrix

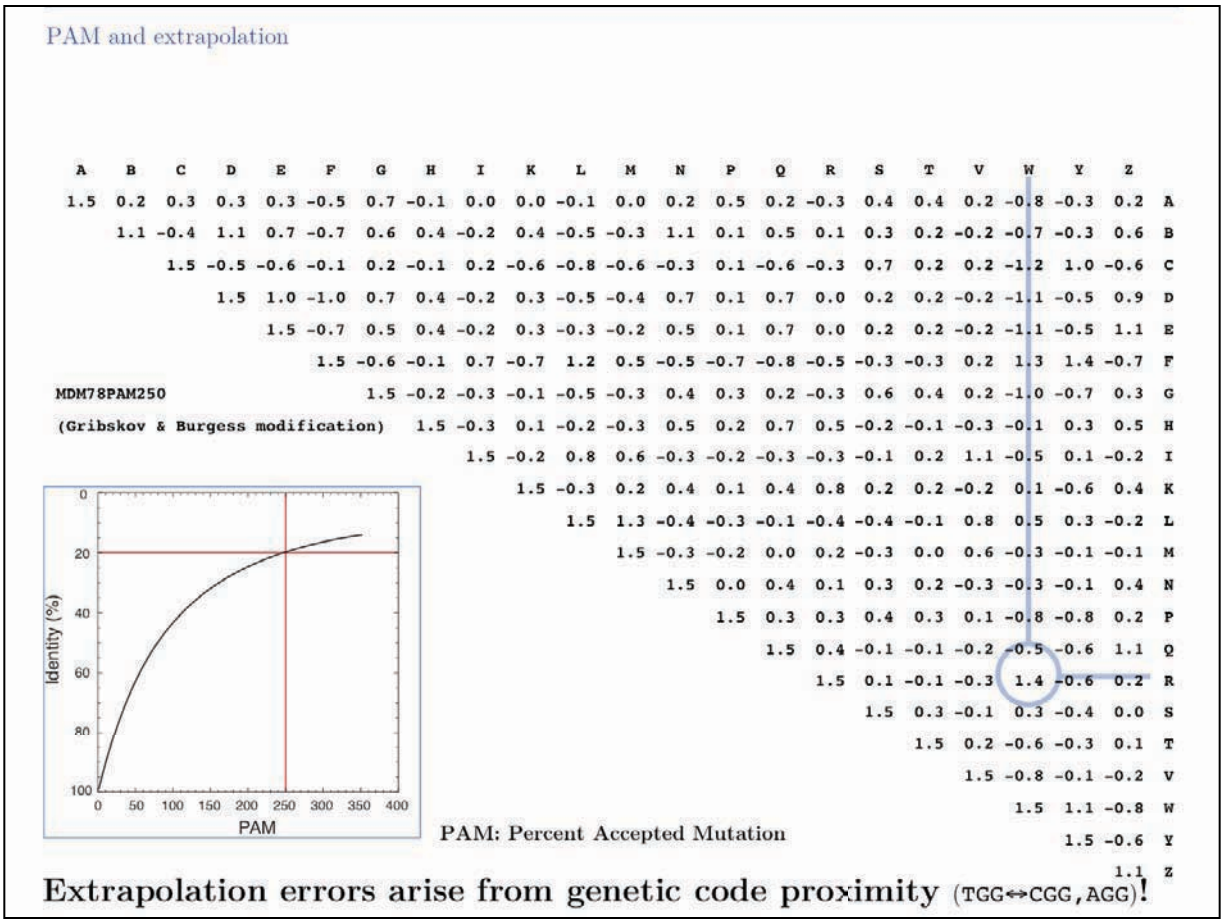
A	B	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y	Z	
1.5	0.2	0.3	0.3	0.3	-0.5	0.7	-0.1	0.0	0.0	-0.1	0.0	0.2	0.5	0.2	-0.3	0.4	0.4	0.2	-0.8	-0.3	0.2	A
	1.1	-0.4	1.1	0.7	-0.7	0.6	0.4	-0.2	0.4	-0.5	-0.3	1.1	0.1	0.5	0.1	0.3	0.2	-0.2	-0.7	-0.3	0.6	B
		1.5	-0.5	-0.6	-0.1	0.2	-0.1	0.2	-0.6	-0.8	-0.6	-0.3	0.1	-0.6	-0.3	0.7	0.2	0.2	-1.2	1.0	-0.6	C
			1.5	1.0	-1.0	0.7	0.4	-0.2	0.3	-0.5	-0.4	0.7	0.1	0.7	0.0	0.2	0.2	-0.2	-1.1	-0.5	0.9	D
				1.5	-0.7	0.5	0.4	-0.2	0.3	-0.3	-0.2	0.5	0.1	0.7	0.0	0.2	0.2	-0.2	-1.1	-0.5	1.1	E
					1.5	-0.6	-0.1	0.7	-0.7	1.2	0.5	-0.5	-0.7	-0.8	-0.5	-0.3	-0.3	0.2	1.3	1.4	-0.7	F
						1.5	-0.2	-0.3	-0.1	-0.5	-0.3	0.4	0.3	0.2	-0.3	0.6	0.4	0.2	-1.0	-0.7	0.3	G
							1.5	-0.3	0.1	-0.2	-0.3	0.5	0.2	0.7	0.5	-0.2	-0.1	-0.3	-0.1	0.3	0.5	H
								1.5	-0.2	0.8	0.6	-0.3	-0.2	-0.3	-0.3	-0.1	0.2	1.1	-0.5	0.1	-0.2	I
									1.5	-0.3	0.2	0.4	0.1	0.4	0.8	0.2	0.2	-0.2	0.1	-0.6	0.4	K
										1.5	1.3	-0.4	-0.3	-0.1	-0.4	-0.4	-0.1	0.8	0.5	0.3	-0.2	L
MDM78PAM250											1.5	-0.3	-0.2	0.0	0.2	-0.3	0.0	0.6	-0.3	-0.1	-0.1	M
(Gribskov & Burgess modification)												1.5	0.0	0.4	0.1	0.3	0.2	-0.3	-0.3	-0.1	0.4	N
													1.5	0.3	0.3	0.4	0.3	0.1	-0.8	-0.8	0.2	P
														1.5	0.4	-0.1	-0.1	-0.2	-0.5	-0.6	1.1	Q
															1.5	0.1	-0.1	-0.3	1.4	-0.6	0.2	R
																1.5	0.3	-0.1	0.3	-0.4	0.0	S
																	1.5	0.2	-0.6	-0.3	0.1	T
																		1.5	-0.8	-0.1	-0.2	V
																			1.5	1.1	-0.8	W
																				1.5	-0.6	Y
																					1.1	Z

A scoring matrix is a tool to quantify how well a certain model is represented in two aligned sequences. The Dayhoff Matrix measures the likelihood that **one amino acid could have been selected by evolution as an acceptable change in closely related sequences.**

MDM78PAM250 is a frequently used mutation data matrix. It is the Margret Dayhoff Model of 1978, extrapolated to a Percent Accepted Mutation rate of 250.

But the matrix as used in many alignment tools does not actually give the original numbers: it has been modified to score all identities the same (i.e. 1.5, which is IMO a big source of alignment problems), and it has been abbreviated to easily map to integers – both changes were done to speed up computation which was a big concern at the time these matrices were written.

This approach has been superseded.

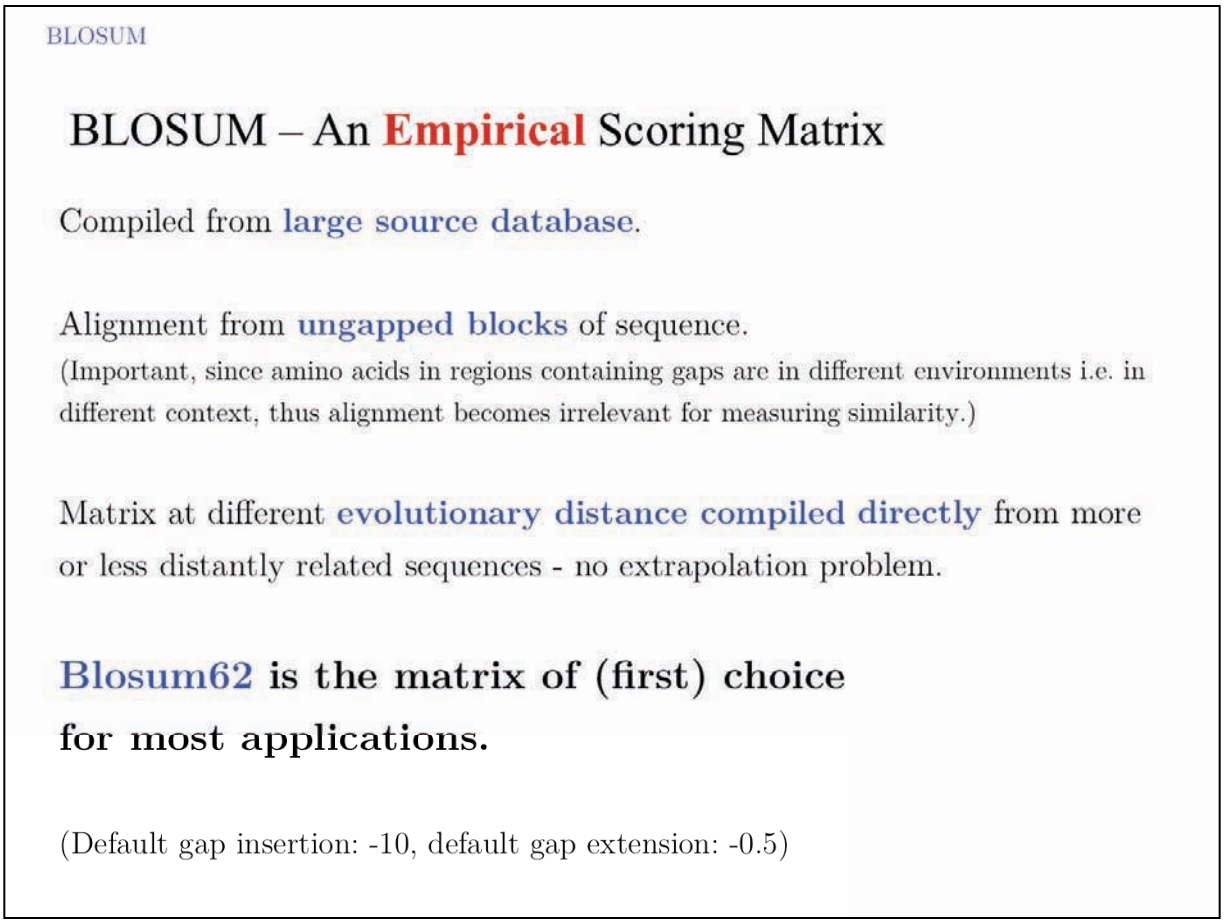


PAM 250 means: 250 accepted changes in the evolution of 100 amino acids of sequence: Percent Accepted Mutations. It expresses the evolutionary distance for which the matrix best describes the likelihood of relatedness. But how can the value of Percent Accepted Mutations be more than 100?

Mutations are located randomly in the sequence, therefore some amino acids may be hit several times and others never at all. Moreover, once an amino acid is changed, it may still revert to its original state through a second mutation. It is easy to see that even with very, very many mutations it is virtually impossible to arrive at a sequence that is 100% different from the original sequence.

As the graph inset shows, PAM250 corresponds to about sequence 20% identity.

Extrapolation to large PAM distances has problems. For example, since Arg and Trp have similar codons (_GG), an R→W mutation is quite likely at the very close evolutionary distances of the proteins in the Dayhoff dataset. It is also quite likely that evolution will favor secondary mutations at that site, to introduce an amino acid that is biophysically more compatible, and theR→W becomes unlikely in more distantly related pairs. But in the Dayhoff model, where large evolutionary distances are extrapolated by repeatedly multiplying the matrix with itself, that discrepancy gets amplified and as a result the pairscore of R→W is almost as high as an identity.



To address the extrapolation problem, Steve Henikoff compiled matrices directly from blocks of ungapped alignments of sequences at given evolutionary distances, once a sufficient number of such sequences were available in the databases. These are the BLOSUM matrices.

BLOSUM62 is a matrix compiled from sequences of not more than 62% identity. It corresponds approximately to a PAM160 matrix and appears to be the most sensitive choice to search for just barely detectably related sequence pairs.

Henikoff, S.; Henikoff, J.G. (1992). Amino Acid Substitution Matrices from Protein Blocks. PNAS **89**:10915–10919.

Eddy, S: (2004), *Nat Biotechnol.* **8**:1035-1036

See also: http://en.wikipedia.org/wiki/BLOSUM (Good article!)

BLOSUM

BLOSUM 62 is a matrix calculated from blocks of aligned sequences with no less than 62% divergence.

A scoring matrix is a tool to quantify how well a certain model is represented in two aligned sequences. The BLOSUM Matrix measures the likelihood that **one amino acid could replace another in ungapped regions of two distantly related sequences.**

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

Henikoff S & Henikoff JG (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA* **89**:10915-10919

Note that the R→W pairscore of BLOSUM62 is very much more in line with our biological intuition.

The matrix has been scaled to integers, for ease of computation. Also, its overall expectation value is negative, so we can't increase alignment scores by randomly adding pairs. This is important for *local alignments*. Finally, as we would expect, the score of residue identities depends on the nature of the residue: e.g. C, H, or W identities are (and should be) more significant than A or L.

Dotplot

Dotplot: simplest approach to sequence comparison:

Good for quick, graphical overview:

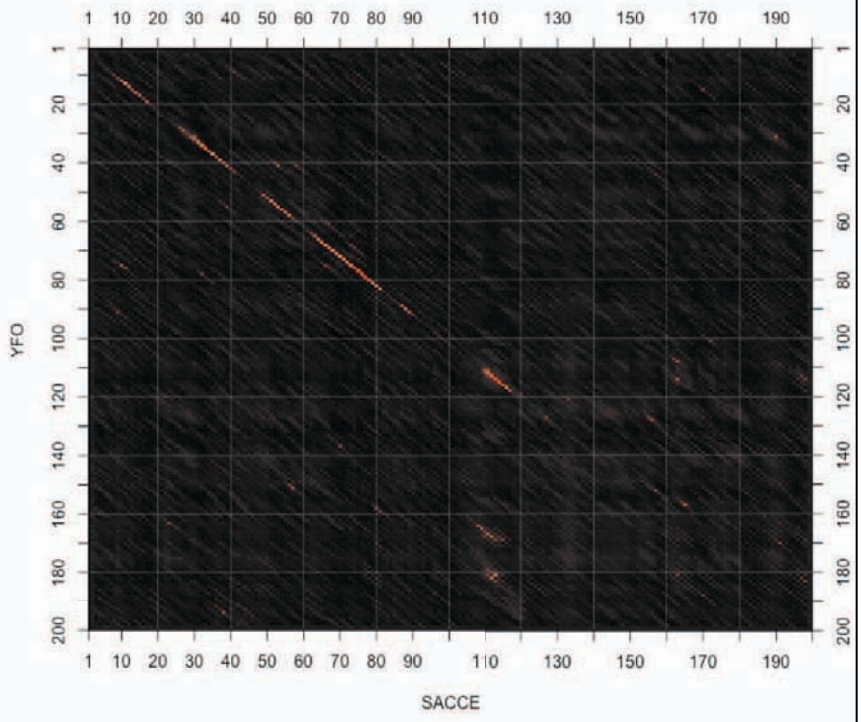
- Inter- and intra- sequence comparison
- Identifies alternative alignments
- Identifies internal repeats and domains
- Identifies low-complexity regions
- Identifies palindromes
- Identifies frameshifts and longer insertions

The simplest approach to sequence comparison is to display scores or identities in a matrix that has one sequence along its rows, one sequence along its columns, thus each cell corresponds to a pairwise comparison.

Dotplot example

Dotplot coded in R

(Comparison of the N-termini of the homologous SACCE and USTMA Mbp1 sequences)



Computing such plots is part of assignment 4.

section

Optimal Alignment

How do we prove homology ?

If the alignment of two sequences scores so highly under a particular model of evolution from a common ancestor that a random chance similarity is sufficiently improbable, we may assume the sequences to be homologous.

How do we measure compatibility with a model of evolution ?

Use a Scoring Matrix that quantifies relatedness *under a model of evolutionary relatedness*. Then score the correct alignment.

What is the "correct" alignment ?

That is an alignment that pairs up those and only those residues that are the result of divergent evolution from a common ancestor.

How do we generate the "correct" alignment ?

We can't. We can never guarantee that a particular alignment is correct! There is no possibility to *know* the ancestral sequence and the evolutionary sequence. Even the sequencing of ancient DNA does not guarantee we are looking at the actual progenitors.

What can we do ?

We can produce an optimal alignment. If the optimal alignment does not support homology, then the correct alignment will not support homology either. But: we cannot guarantee that this is the correct alignment.

(In fact we can define scenarios in which it will not be, since a one-to-one relationship between residues may not be meaningful in distantly related sequences.)

In the absence of observation, the correct alignment remains unknown.
However: ...

If we produce the **best possible** alignment and we cannot infer homology from that, the "*correct*" alignment would not convince us either.

... and the *best possible alignment* **can** be constructed.

optimal alignment

How can the best possible alignment be constructed ?

Can one generate all alignments, score them, and chose the best ?

Note that *best* in this context means: highest scoring.

... No. The existence of *indels* makes it intractable to consider all possible alignments.

indels

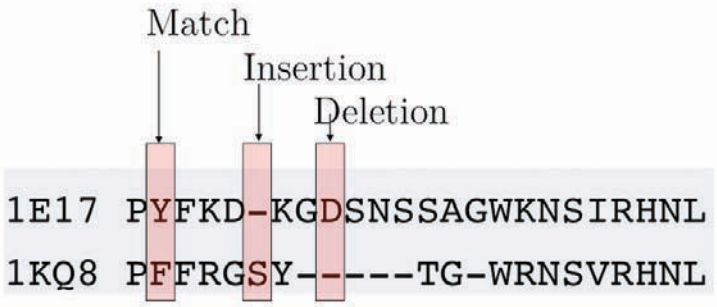
Related sequences often have different lengths. Ends can be lengthened and shortened, and internally, segments ranging from single residues to entire domains can have been inserted.

```
1E17 PYFKD-KGDSNSSAGWKNSIRHNL
1KQ8 PFFRGSY-----TG-WRNSVRHNL
```

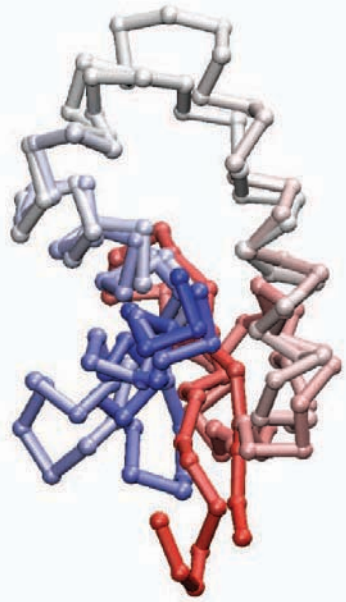
In general, an insertion from the point of view of one sequence is the same as a deletion from the point of view of the other sequence, thus we often use the term "**indel**".

Note that the term insertion or deletion refers only to the sequences, not to the actual molecular event!

indels



Since every position of the alignment can represent one of three states, the number of different alignments is on the order of (3^{length})–greater than the number of particles in the universe for the length of typical protein sequences.
This is an *intractable* problem.



Number of particles in the universe: on the order of 10^{81}.

Alignments for two sequences of length 200: $\sim 3^{200} = 10^{95}$.

indels

But: if we assume that the *global score is simply a sum of pair-scores*, we can devise an effective divide-and-conquer approach ...

1E17 RHNLSLHSKFIKVH-N--EATG---KSS-WWMLNPEGG
1KQ8 RHNLSLN-DCFVKVLRDPSR-PWGKDNYWMLN-P----

optimal alignment

Premise:

The total alignment score is the sum of all pair-scores for characters, minus a penalty for each indel.

Pair scores depend only on the pair of characters under consideration.

Since we determine a pair-score "locally" (without reference to its neighbourhood, or other context), simply by *looking it up in a scoring matrix*, we can subdivide the big problem of global alignment into many little problems that are easier to solve.

The premise of context independence makes finding an optimal alignment a solvable problem. It is can be shown that alignment problems that are not context-independent are NP hard, i.e. no algorithm exists that solves such a problem in a number of steps that is proportional to some polynomial of the alignment length. Rather, the number of steps in fully context-sensitive, gapped alignment must be proportional to some number to the power of the alignment length.

You can visualize this by considering that *context sensitive* really means: each local decision (whether to match two characters or insert an indel) is influenced by the state of all characters already in the alignment: all combinations of states are therefore distinct and must be considered separately. This is exactly the procedure which we have considered previously as the *brute-force* approacht o constructing alignments – and found to be intractable.

optimal alignment

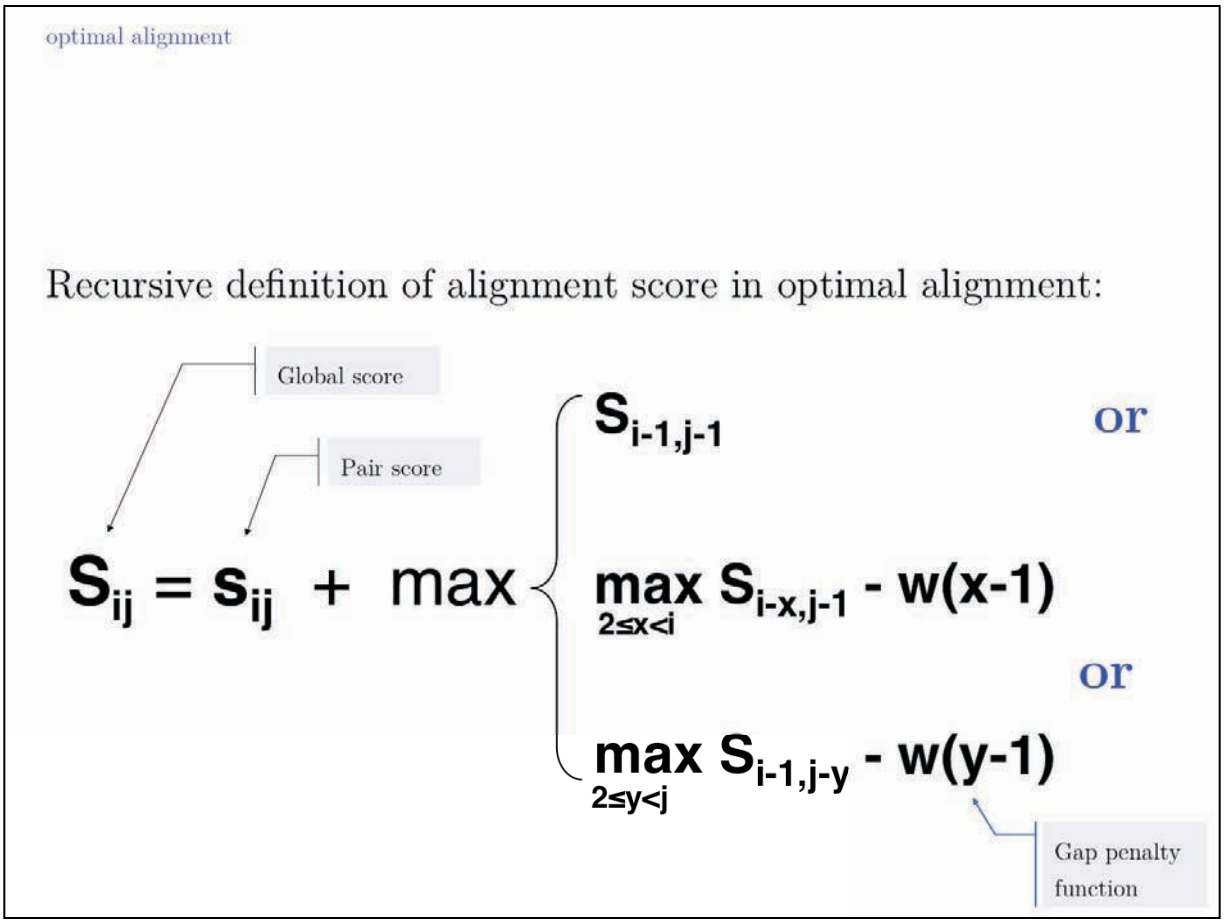
The highest possible score of an alignment is the **(highest possible score of an alignment that is one residue shorter)**, extended in the best possible way by one residue ...

... the highest possible score of an alignment that is one residue shorter is the **(highest possible score of an alignment that is two residues shorter)**, extended in the best possible way by one residue ....

... the highest possible score of an alignment that is two residues shorter is the **(highest possible score of an alignment that is three residues shorter)**, extended in the best possible way by one residue ....

... the highest possible score of an alignment that is three residues shorter is the (highest possible score of an alignment that is four residues shorter), extended in the best possible way by one residue ...

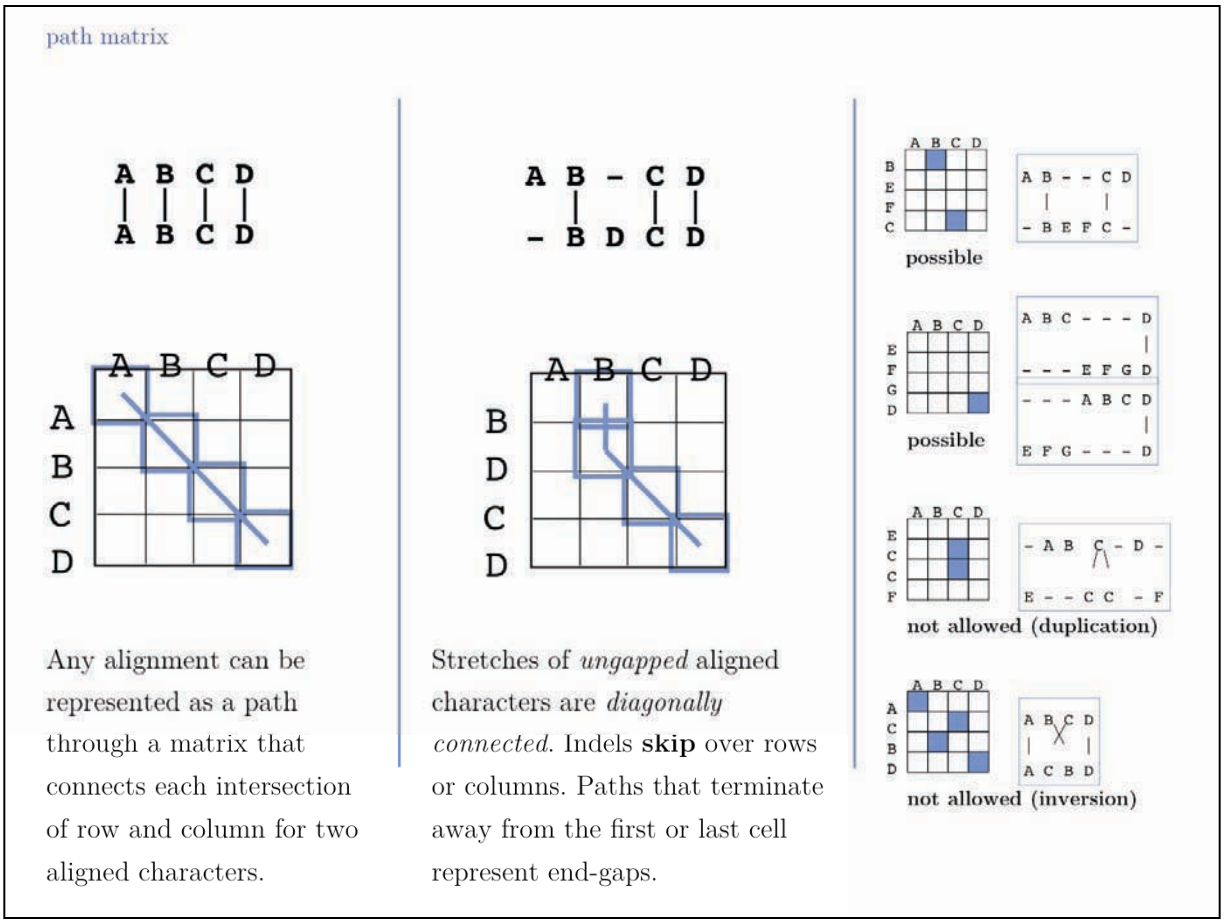
... the highest possible score of an alignment that contains only a single pair of residues can be looked up in the scoring matrix.



Optimal alignment, in the way we have defined the procedure a few slides ago, is simple to write as a recursion. However, implementing the approach as a recursion is very(!) inefficient since it requires looking up many values over and over again. For example if we are to calculate the score for i=9, j=10, we need to consider as one one of the possible extensions the cell i=8, j=9 and x=4 i.e. we need to calculate $s_{8\text{-}4,9\text{-}1}\text{-}w_{4\text{-}1} = s_{4,8}\text{-}w_3$. But this is the same value for s we previously had to calculate for the adjacent cell column: i=7, j=9, x=3: $s_{7\text{-}3,9\text{-}1}\text{-}w_{3\text{-}1} = s_{4,8}\text{-}w_2$, only with a different w. It is not the w-values that are costly to calculate however, but the s-values themselves, since we need to recurse all the way to the Base Case each time we want to calculate one. So while it is compact to write the alignment in the way given above, in practice we would want to store each intermediate result that is going to be reused. This technique of storing useful intermediate results is called **Memoization** (not memo r ization) in computer science.

(cf. http://en.wikipedia.org/wiki/Memoization)

The actual algorithm therefore uses a compact and intuitive way to model the problem: store intermediate values in a matrix where rows and columns correspond to characters in the respective sequences. The highest score in the matrix is the optimal score and the cells that contribute to that score define the optimal alignment.



An alignment can be represented as a **path** through a **matrix** that has a row resp. column for every letter of the two sequences to be aligned. Any alignment can be represented as a path in such a matrix. Only a subset of arrangements correspond to legal paths that represent our normal definition of an alignment.

Note that – especially in genome/genome comparisons – duplications and inversions are common and specialized algorithms are available to perform such alignments (e.g. Shuffle-LAGAN (http://lagan.stanford.edu/)).

algorithm

Needleman & Wunsch (1970):
the optimal alignment is given by the path that leads to the highest possible sum of all the pair-scores it contains.

First step: compile all pairwise scores into a matrix.

	A	B	C	D
B	0	**1**	0	0
D	0	0	0	**1**
C	0	0	**1**	0
D	0	0	0	**1**

(This example assumes: identity matrix - match=1, mismatch=0, no gap penalties)

The first step of the Needleman-Wunsch algorithm for global, optimal sequence alignment. This algorithmic strategy is frequently referred to as *Dynamic Programming*.

- http://en.wikipedia.org/wiki/Dynamic_programming
- http://en.wikipedia.org/wiki/Needleman-Wunsch_algorithm
- http://www.avatar.se/molbioinfo2001/dynprog/dynamic.html – Dynamic programming example, courtesy of Per Kraulis.

algorithm

Second step: The highest score in the last column and row is the is highest pairscore we put there from the scoring matrix. This is the Base Case, if we think about the recursion, because there is no previous score we had to consider.

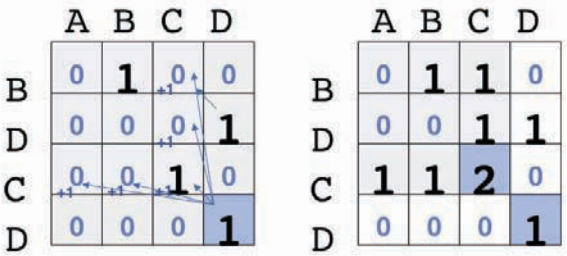
	A	B	C	D
B	0	**1**	0	0
D	0	0	0	**1**
C	0	0	**1**	0
D	0	0	0	**1**

(This example assumes: identity matrix - match=1, mismatch=0, no gap penalties)

The next scores we need to calculate are the cells in the previous column or row...

algorithm

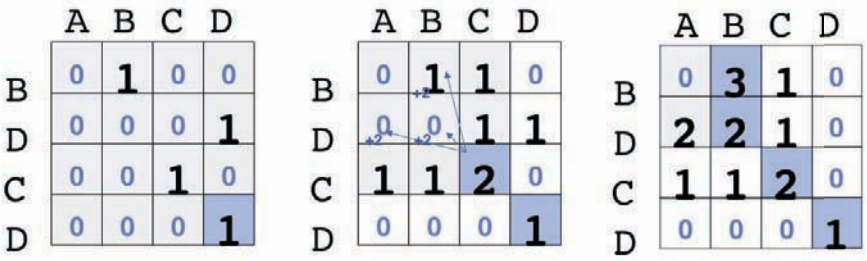
Third step: Extend the path. Assign to each cell of the next column and row the highest value we can get by adding to its current value a value from a previous cell **that could be part of an alignment path**.



(This example assumes: identity matrix - match=1, mismatch=0, no gap penalties)

algorithm

Repeat: (Assign to each cell of the next column and row the highest value we can get by adding to its current value a value from a previous cell **that could be part of an alignment path**.)

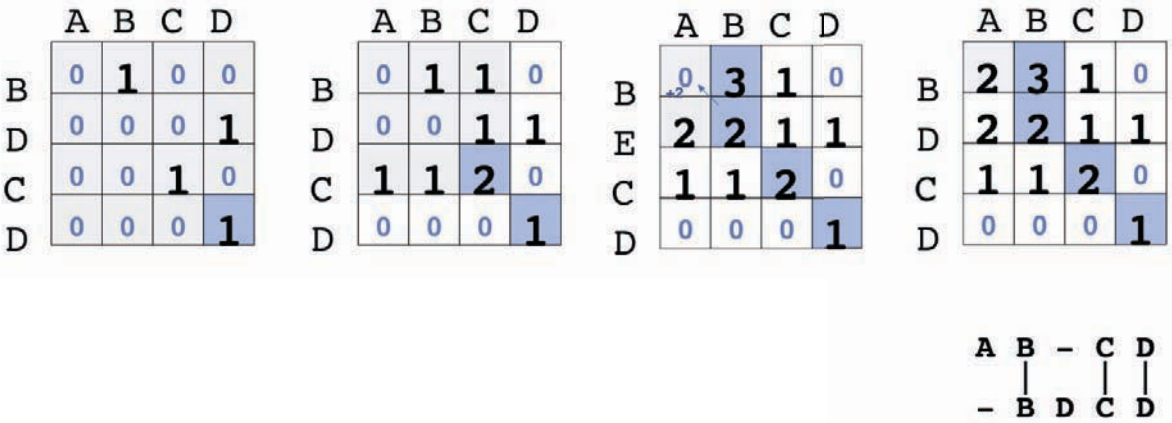


(This example assumes: identity matrix - match=1, mismatch=0, no gap penalties)

algorithm

Final step: The **highest possible** score for the alignment path matrix is found after the matrix is filled.

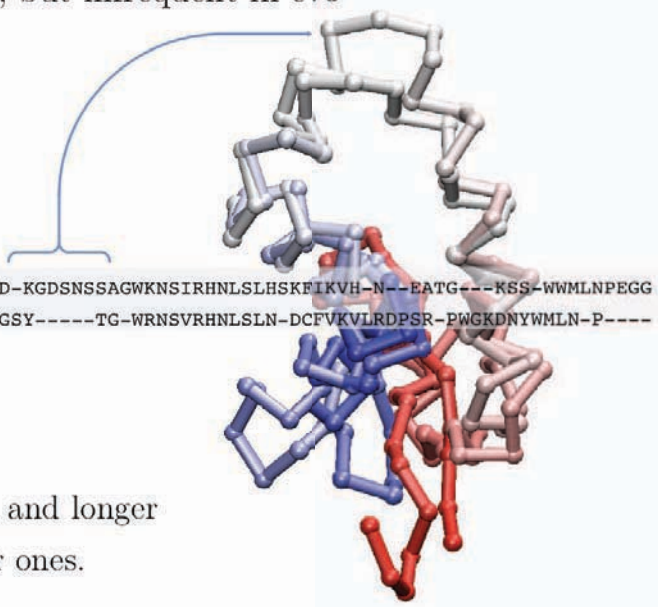
Once the highest possible score has been determined, we only need to find the **cells that have contributed to this score**. The optimal alignment is given by the path that contains these cells. The cells are simply retrieved by backtracking.



(This example assumes: identity matrix - match=1, mismatch=0, no gap penalties)

indels

In reality, related sequences *often* have different lengths. Ends can be lengthened and shortened, and segments ranging from single residues to entire domains can have been inserted or deleted. We need to take into account that indels are possible, but infrequent in evolution.



```
1E17 YAELISQAIESAPEK-RLTLAQIYEWMVRTVPYFKD-KGDSNSSAGWKNSIRHNLSLHSKFIKVH-N--EATG---KSS-WWMLNPEGG
1KQ8 YIALITMAIR-DSAGGRLTLAEINEYLMGKFPFFRGSY-----TG-WRNSVRHNLSLN-DCFVKVLRDPSR-PWGKDNYWMLN-P----
```

Empirically:
Insertion/deletion events are rare, and longer gaps are less frequent than shorter ones.

... unfortunately, we have no quantitative, mechanistic model for these events.

Commonly, a gap penalty is calculated from a constant value for opening the gap (to reflect the rarity of the event) and an increment for every extension (to reflect the fact that longer gaps are less frequent than shorter ones).

$$w(l) = a + bl$$

This type of gap penalty is called an **affine gap model**.

It does not reflect exactly what we actually observe in biology.

scoring indels

Database analysis shows that gaps are log-distributed.

An attempt to model this situation has proposed a sum of exponentials ...

$$P(n) = \sum_i A_i e^{n\lambda_i}$$

... but other studies have **not** shown a clear advantage of logarithmic over affine gap penalties.

Qian & Goldstein (2001)
Proteins B:102-104

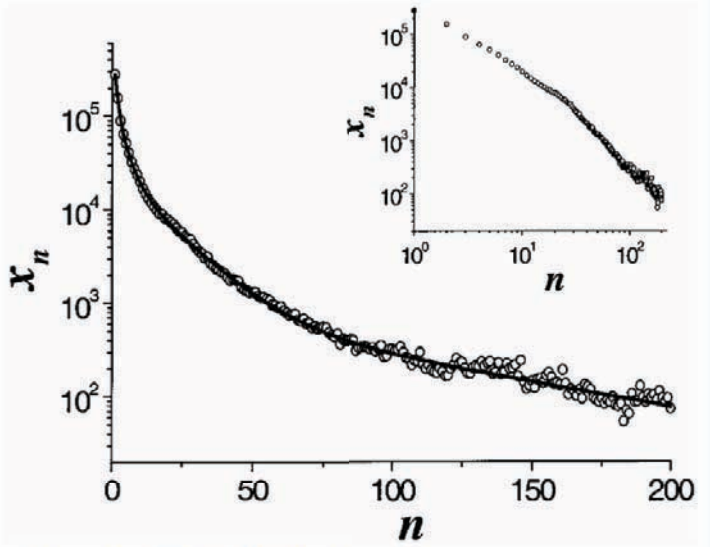


Fig. 1. Log plot and log-log plot of observed structure-based gap length distribution, compared with a quadruple-exponential fit (Eq. 8).

Note that discrete slopes in the log-log plot may indicate discrete molecular mechanisms for short- and long indels with characteristic length/frequency relationships.

Qian and Goldstein (2001)[1] have shown that a log linear plot of gap probabilities in aligned sequences can be modeled by a sum of four exponential functions. This can be interpreted to mean that several molecular mechanisms could exist for the generation of indels, each with a distinct and characteristic probability of occurrence.

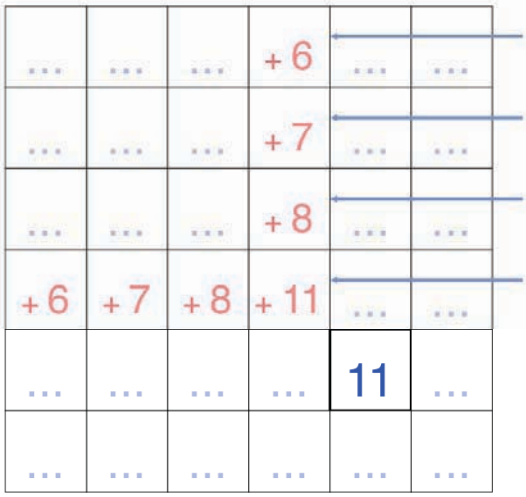
However, logarithmic gap penalties do not improve alignments (Cartwright, 2006)[2]. Recent developments focus on the inclusion of additional knowledge about the sequences, such as secondary-structure specific gap penalties, or using sequence profiles or multiple alignments, rather than aiming to further improve the gap parameters. The bottom line is: we have no good model for indels, but we have no significantly better model than the simple affine model.

[1] http://www.ncbi.nlm.nih.gov/pubmed/11536366

[2] http://www.ncbi.nlm.nih.gov/pubmed/17147805

indels

Calculating **affine gap penalties** is computationally simple:
reduce the score that is added to a cell according to the number of rows or columns that need to be skipped.



Example parameters:

Gap insertion: -3

Gap extension: -1

Insert a gap,
extend by two: add score – 3 – 2

Insert a gap,
extend by one: add score – 3 – 1

Insert a gap: add score – 3

Ungapped continuation: add score to cel

Pairwise Optimal alignment

- Reasonably fast for pairwise gene comparisons.
- Too slow / needs too much memory for database scans or whole genome alignments.
- Guaranteed to always give a mathematically optimal alignment.
- Alignment not guaranteed to be biologically correct or unique.
- Alignment will depend on scoring matrix.
- Alignment will strongly depend on (empirical !) gap insertion and extension parameters.

local and global ...

Often the score for an alignment between two substrings can be larger than the score for an alignment between two entire sequences. This is especially the case if a sequence has several domains.

The Smith-Waterman variation of the Needleman-Wunsch algorithm computes the highest scoring aligned *substrings*.

Always use local alignment -

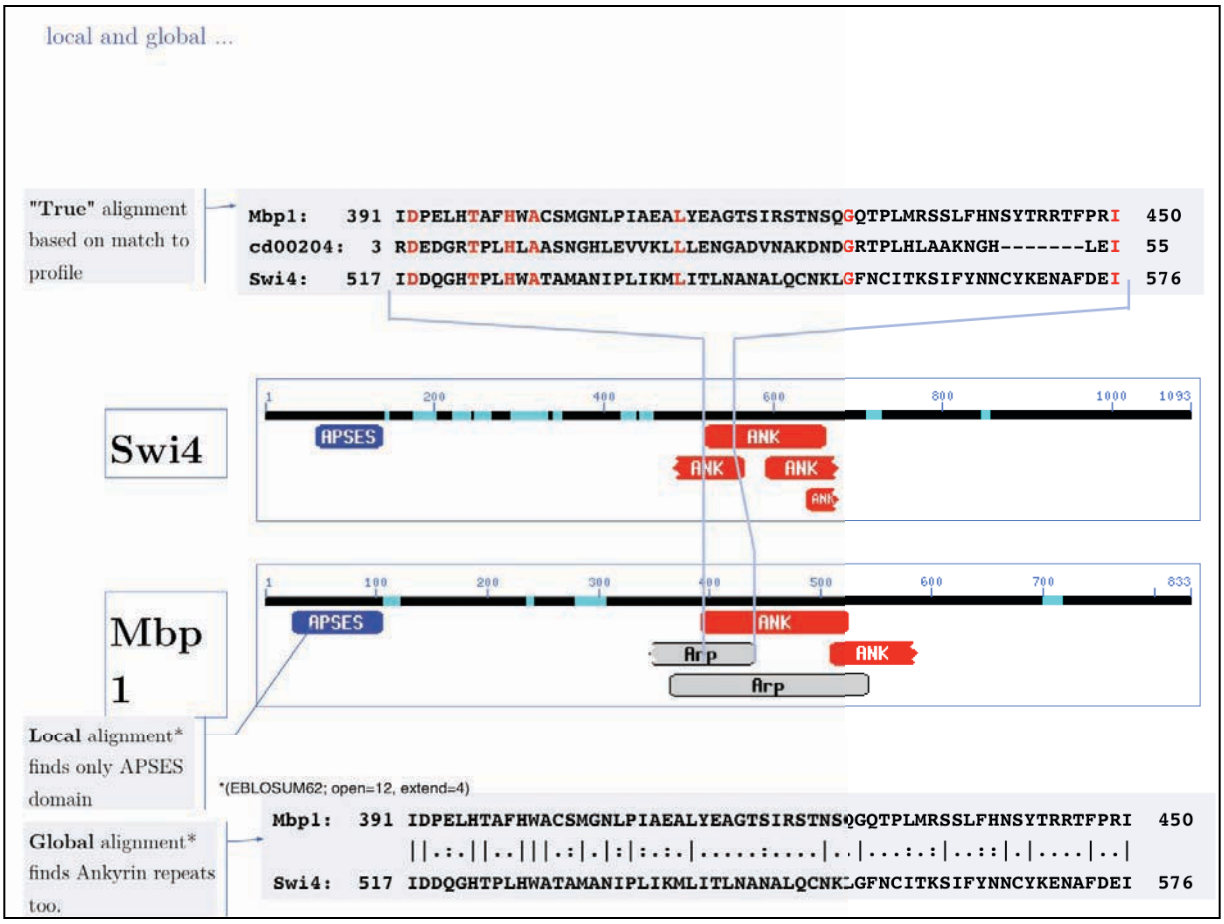
- when the sequences have very different lengths
- when the sequences are only related in domains or subdomains

In the example above, the ankyrin domain repeats of the yeast transcription factor Mbp1 are shown as a red box in this graphic of domains in sequence families, compiled in the **CDART database**[1]. This domain is found in many other proteins, but some of them do not share the other sequence elements found in Mbp1 - they are only partially related. Attempting a global sequence alignment with such sequences would attempt to align sequences that are actually not homologous, leading to inappropriately low scores and the danger of spurios results.

Temple Smith and Michael Waterman[2] have slightly modified the Needleman-Wunsch algorithm, 11 years after its publication, to find the highest scoring **local** alignment: this is the highest match in the matrix, tracked back to the point where the pathscore drops below zero. The rest of the algorithm works in exactly the same way. There is only one detail that needs to be considered: the substitution matrix must yield a negative expectation value for random alignments. If this were not the case, random pairs could extend the locally high-scoring alignment unreasonably.

[1] http://www.ncbi.nlm.nih.gov/pubmed/12368255

[2] http://en.wikipedia.org/wiki/Smith_Waterman_algorithm



In this example, Smith-Waterman local sequence alignment detects only the high-scoring similarity between Mbp1 and Swi4s APSES domains. The lower scoring, more highly diverged ankyrin repeats are missed by the algorithm. The Needleman-Wunsch alignment finds both sets of sequences, albeit there are segments in between that don't align well at all.

In this case, one **could** do a local alignment, remove the matching segments from the input sequences and then redo the alignment to see if any other significantly similar segments are found.

When to use ...

No alignment ...

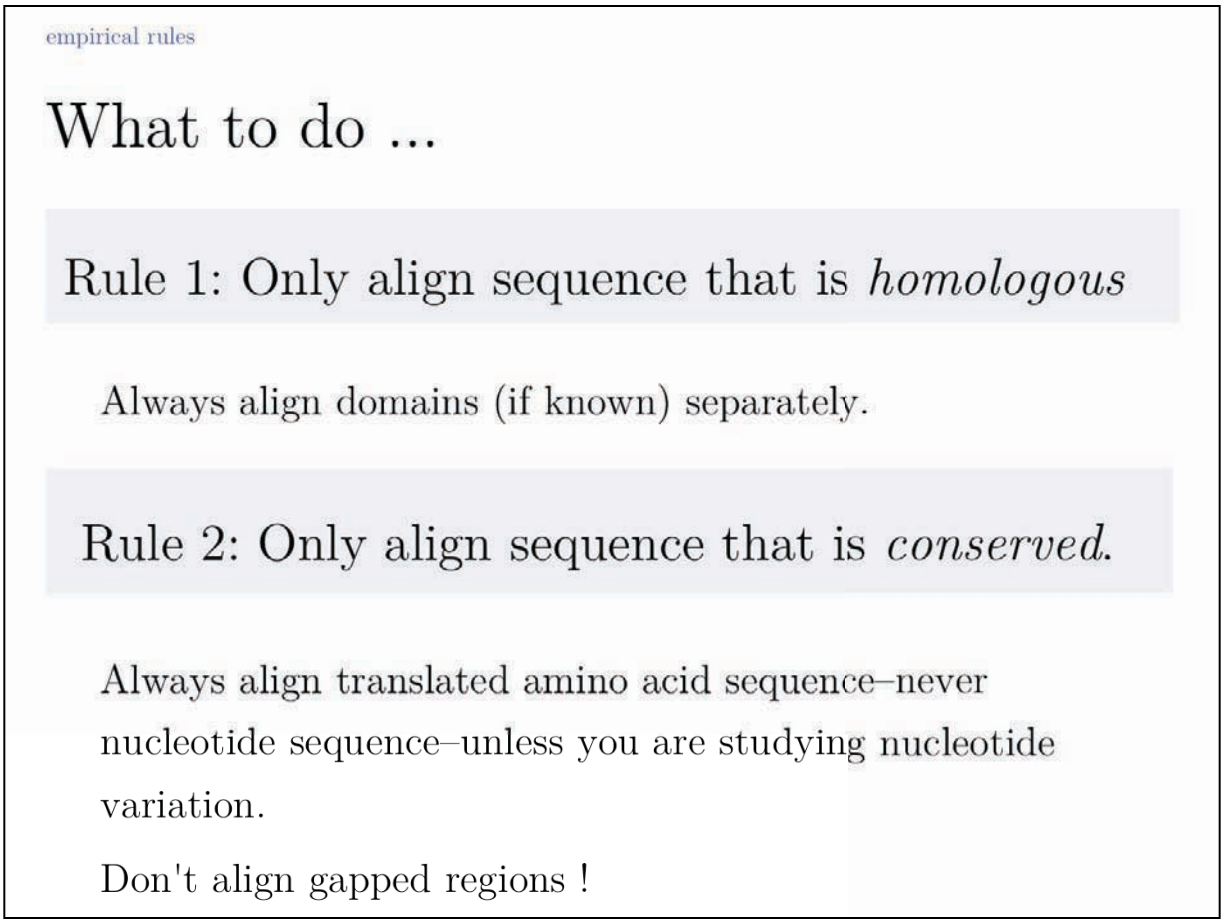
Annotations of functional elements or domains may be conserved (e.g. TM-helices, phosphorylation sites, 2° structure, disordered segments ...). Especially significant if sequence divergence is otherwise large.

Local alignment ...

Alignment in parts. Appropriate if sequences are homologous only in part, or if parts of the sequence are structurally dissimilar, or if inserted domains would create unrealistically large gap penalties. May need to be iterated.

Global alignment ...

Appropriate if sequences are homologous over their whole length, especially to bridge segments of high divergence, and to discover islands of high similarity.



Of course the algorithms will optimally align anything you feed them, but for anything but homologous sequence **the alignment will be meaningless**. Aligning non-homologous sequences is a nice example of cargo-cult bioinformatics.

Therefore: if you already know that your proteins are multi-domain, separate out the domains before aligning. If you don't know, critically look at the results, generate a hypothesis about the domain structure and rerun your alignment on the domains separately. The exception, of course: is if you know (or believe) your two proteins comprise homologous domains in the same order.

Amino acid sequences are much more highly conserved then genomic sequence and even if you have nucleotide sequences to start from, you should always **translate** them before aligning. In general, many more matches are required to make nucleotide sequence matches significant, since the alphabet is much smaller. Also, there is no good notion of "similarity" or "conservative mutation" at the nucleotide level[1].

The only reasons to align nucleotides are:

- if you are actually interested in the **number and type of nucleotide exchanges**, such as in gene assembly and EST clustering, studies of SNPs, in comparative genomics, phylogenetic studies of closely related genes, or defining primer binding sites;
- if you are **aligning untranslated sequence**s; in particular if it is the nucleotide sequence itself that is conserved, such as in DNA binding sites or splice sites; or if you are studying RNA genes, such as tRNA or rRNA.

A corollary is that you should not try to align sequences in highly gapped regions. These residues have evolved in a non-comparable context, they cannot have been conserved by evolution for that reason and applying our scoring matrices cannot compare such residues in a meaningful way.

[1] However, transitions (conserving pyrimidines or purines) are more frequent than transversions. See http://en.wikipedia.org/wiki/Models_of_DNA_evolution for how this is modelled.

How to set penalties ...

Higher opening penalties make gaps *less frequent*.

Higher extension penalties make gaps *shorter*.

The effect of the penalties depends on the scoring matrix!

Typical opening penalty: *2-3* times an *identity score*.

Typical extension penalty: $^{1}/_{5}$ to $^{1}/_{10}$ of an *opening penalty*.

Default penalties for BLOSUM62: -11 and -1 at *NCBI* (BLAST)
-10 and -0.5 at *EMBOSS* (Needle, Water)

How to report results ...

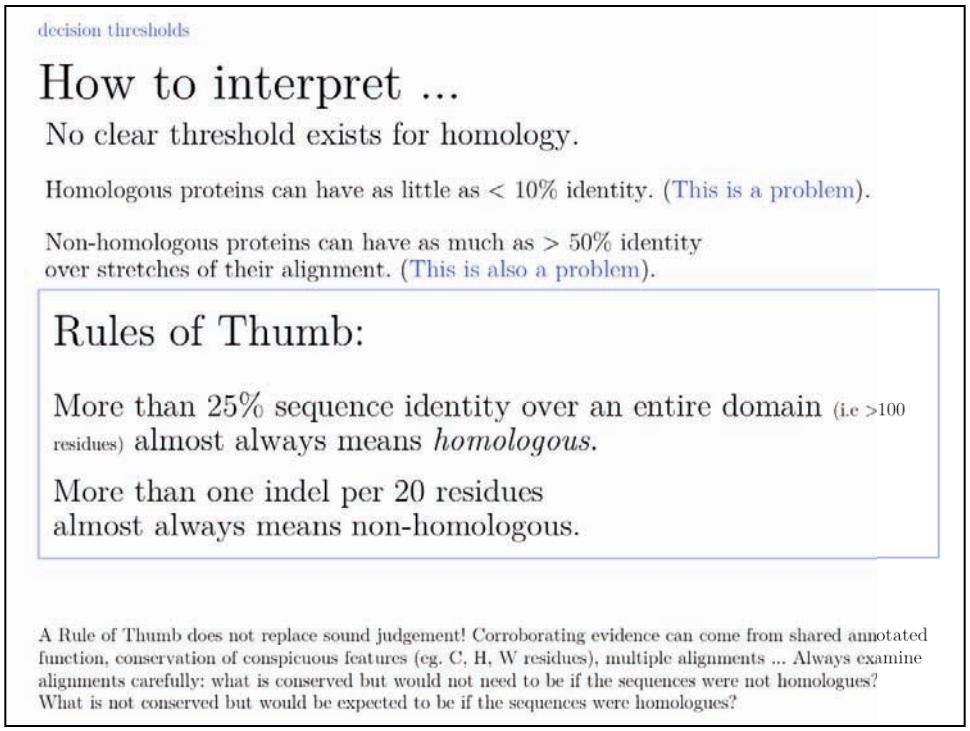
The alignment score is a single number that measures the quality of the alignment. Scores depend on:

- the matrix
- the gap insertion penalty
- the gap extension penalty
- the end-gap penalty
- the algorithm (local or global, optimal or heuristic)

Therefore, all these parameters need to be reported along with the alignment (similarity) score, otherwise the number is meaningless.

Alternatively: *report % identity!* This allows a certain degree of comparison between alignments.

Note that reporting %-identity is an objective metric, but it still depends on the exact alignment that has been produced and it does not capture the quality of gaps.



Empirical thresholds to conclude that two sequences are homologous

Identities of 20 to 25% are also called the ''twilight zone'' - in which homology is likely but can't be confidently inferred from sequence similarity alone.

These thresholds are based on sequence similarity after optimal alignment. Additional supporting evidence for homology can be contributed from:

- simlar length;
- similar functional sequence patterns (e.g. cys/his clusters);
- similar number of transmembrane helices;
- similar conservation patterns or conserved motifs;
- similar amino acid frequencies or bias (eg. polyglutamine, polyproline);
- similar patterns of disordered sequence;
- similar structure;
- similar function;
- similar genomic context;
- similar interactors;
- similar subcellular localization;
- [...]

EMBOSS tools

Needle - for optimal global alignments

Water - for optimal local alignments

stretcher - for long sequences: half as fast as NW but only linear to the shorter sequence in memory.

matcher - for long sequences: slower than SW, but only linear to the shorter sequence in memory; also gives suboptimal matches.

supermatcher - rough results for very long sequences; heuristics, based on word matches.

Needle

Output of EMBOSS package global sequence alignment program *Needle* with sequences for yeast Swi4 and Mbp1.
(Default parameters).

The first ~200 amino acids of the alignment are shown, these include the conserved DNA binding domain.

- | - identical amino acids
- : - highly similar pairs
- . - similar pairs.

```
#=======================================
# Aligned_sequences: 2
# 1: Mbp1
# 2: Swi4
# Matrix: EBLOSUM62
# Gap_penalty: 10.0
# Extend_penalty: 0.5
#
# Length: 1147
# Identity:     266/1147 (23.2%)
# Similarity:   414/1147 (36.1%)
# Gaps:         368/1147 (32.1%)
# Score: 640.0
#=======================================

Mbp1     1 ------------------------------MSNQ--IYSARYSGVDVYE-  17
                                         .||.  |..|.||..||||
Swi4     1 MPFDVLISNQKDNTNHQNITPISKSVLLAPHSNHPVIEIATYSETDVYEC  50

Mbp1    18 FIH--STGSIMKRKKDDWVNATHILKAANFAKAKRTRILEKEVLKETHEK  65
           :|.  .|..:|:|.||||:|.|.:.|.|.|:|.|||:|||||.....|||
Swi4    51 YIRGFETKIVMRRTKDDWINITQVFKIAQFSKTKRTKILEKESNDMQHEK 100

Mbp1    66 VQGGFGKYQGTWVPLNIAKQLAEKFSVYDQLKPLFDFTQTDGSASPPPAP 115
           ||||:|::||||:||:.||.|..|:.:.|.:.......|.|.:..||...
Swi4   101 VQGGYGRFQGTWIPLDSAKFLVNKYEIIDPVVNSILTFQFDPNNPPPKRS 150

Mbp1   116 KH---------------HHASKVDRKKAIRSASTSAIMETKRNNKKAEEN 150
           |:                ...:|..|||. .|:||||  .|...|||.::|
Swi4   151 KNSILRKTSPGTKITSPSSYNKTPRKKN-SSSSTSA--TTTAANKKGKKN 197

Mbp1   151 --------------------QFQ---SSKILGNPTAAPRKRGRPVGSTRG 177
                               |||   |..|:.|     ......:.....
Swi4   198 ASINQPNPSPLQNLVFQTPQQFQVNSSMNIMNN-----NDNHTTMNFNND 242
```

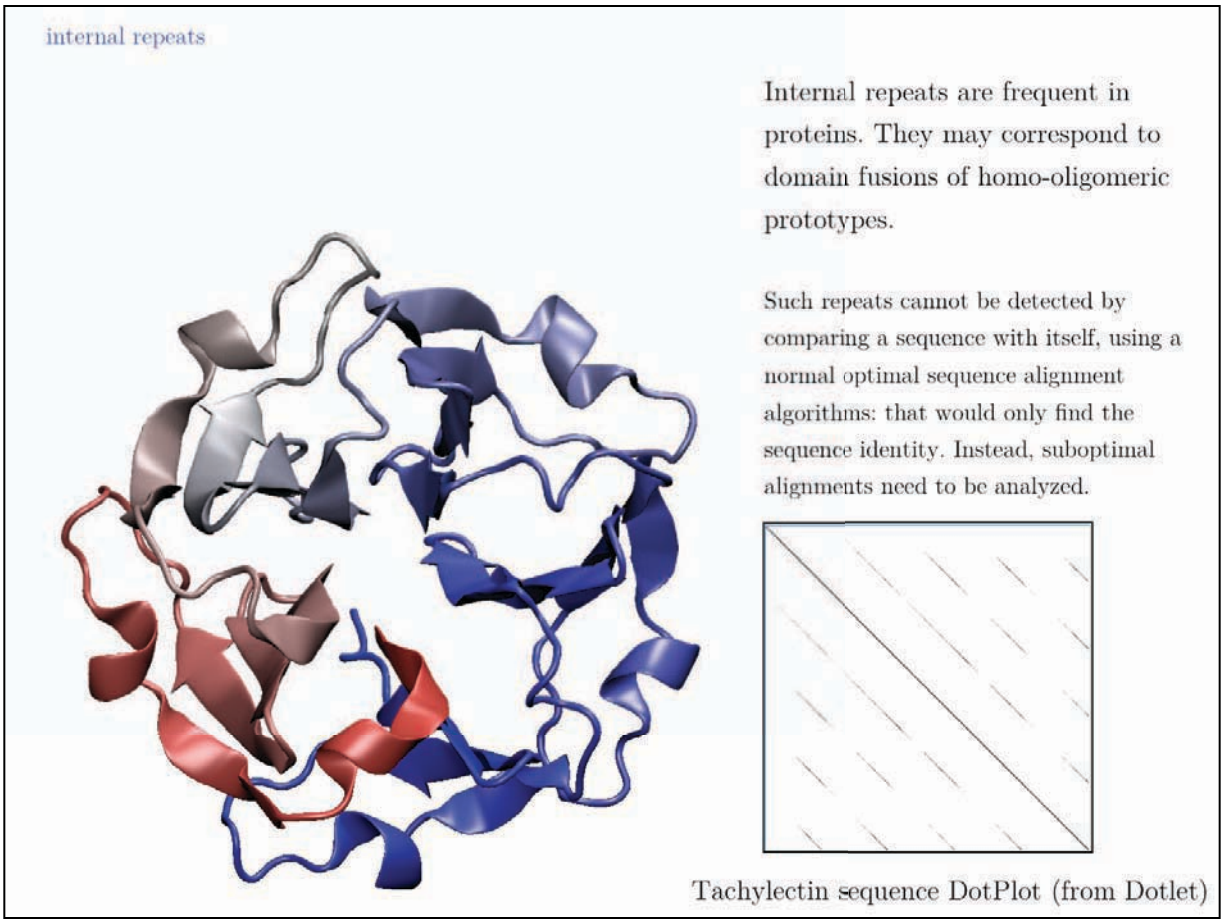
Water

Output of EMBOSS package local sequence alignment program *Water* with sequences for yeast Swi4 and Mbp1.
(Default parameters).

The first ~200 amino acids of the alignment are shown, these include the conserved DNA binding domain.

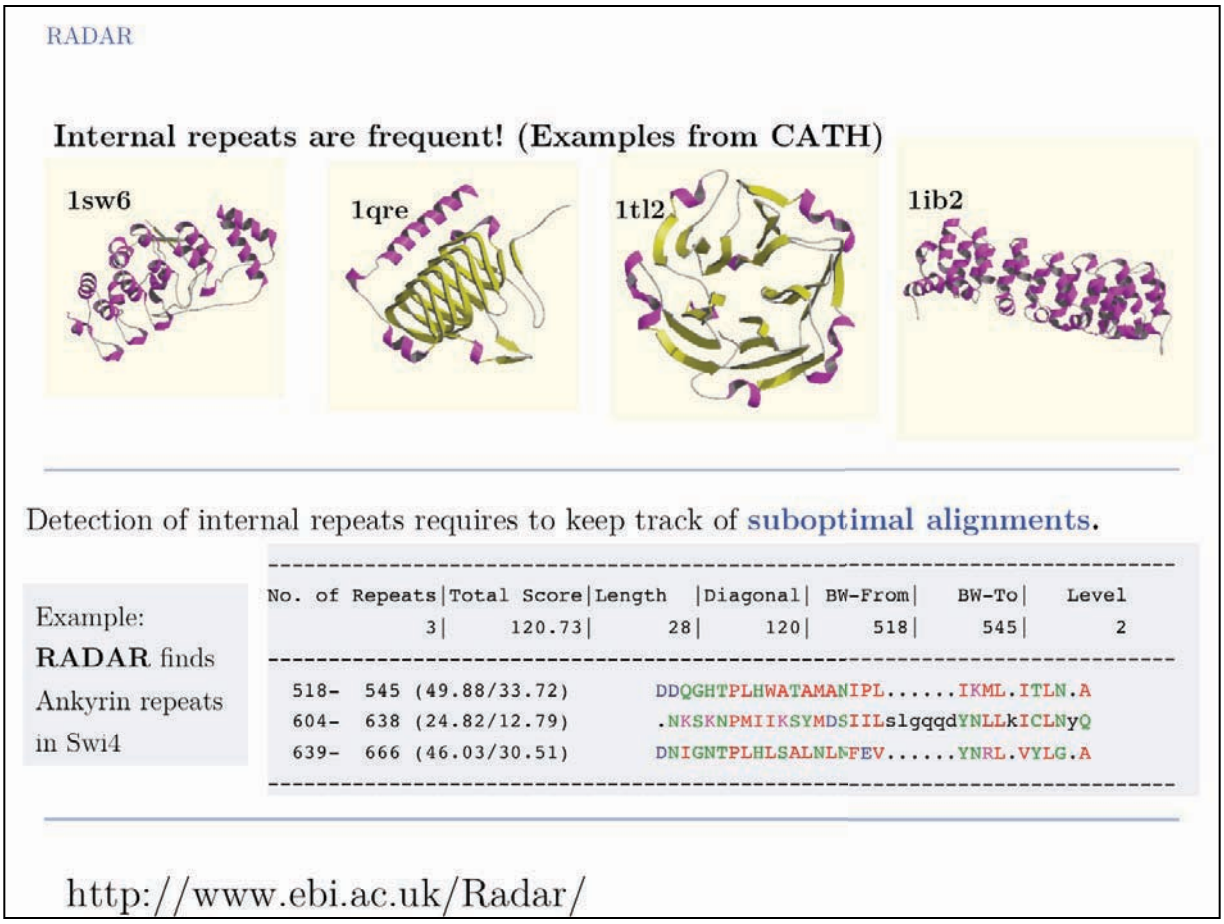
- | - identical amino acids
- : - highly similar pairs
- . - similar pairs.

```
#=======================================
# Aligned_sequences: 2
# 1: Mbp1
# 2: Swi4
# Matrix: EBLOSUM62
# Gap_penalty: 10.0
# Extend_penalty: 0.5
#
# Length: 1088
# Identity:     263/1088 (24.2%)
# Similarity:   411/1088 (37.8%)
# Gaps:         317/1088 (29.1%)
# Score: 644.5
#=======================================

Mbp1      8 ARYSGVDVYE-FIH--STGSIMKRKKDDWVNATHILKAANFAKAKRTRIL  54
            |.||..|||| :|.  .|..:|:|.||||:|.|.:.|.|.|:|.|||:||
Swi4     40 ATYSETDVYECYIRGFETKIVMRRTKDDWINITQVFKIAQFSKTKRTKIL  89

Mbp1     55 EKEVLKETHEKVQGGFGKYQGTWVPLNIAKQLAEKFSVYDQLKPLFDFTQ 104
            |||.....|||||||:|::||||:||:.||.|..|:.:.|.:.......|
Swi4     90 EKESNDMQHEKVQGGYGRFQGTWIPLDSAKFLVNKYEIIDPVVNSILTFQ 139

Mbp1    105 TDGSASPPPAPKH---------------HHASKVDRKKAIRSASTSAIME 139
            .|.:..||...|:               ...:|..|||. .|:||||  .
Swi4    140 FDPNNPPPKRSKNSILRKTSPGTKITSPSSYNKTPRKKN-SSSSTSA--T 186

Mbp1    140 TKRNNKKAEEN--------------------QFQ---SSKILGNPTAAPR 166
            |...|||.::|                    |||   |..|:.|     .
Swi4    187 TTAANKKGKKNASINQPNPSPLQNLVFQTPQQFQVNSSMNIMNN-----N 231

Mbp1    167 KRGRPVGSTRGSRRKLGVNL------------QRSQSDMGFPRPAIPNSS 204
            .....:.....:|..|..|:            |:|..:..|      |::
Swi4    232 DNHTTMNFNNDTRHNLINNISNNSNQSTIIQQQKSIHENSF------NNN 275
```



Tachylectin: cherry-blossom symmetry in a lectin from the Japanese horseshoe crab.

The dotplot that compares the sequence with itself shows the self-identity matches on the diagonal and five domains that are all mutually similar (but not identical) to each other, seen as partial similarities on the off-diagonals. If you think of the path matrix as resembling a dotplot, the repeat alignments we need to analyze correspond to such off-diagonal stretches of high similarity.



The RADAR server[1] at the EBI analyzes sequences for internal repeats.

[1] http://www.ebi.ac.uk/Radar/

summary

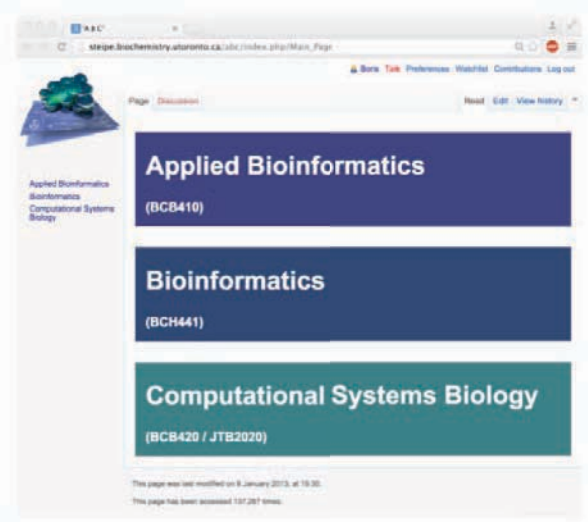
Homologous sequences are similar.

Suitable pair-score matrices can measure similarity under a model of evolutionary conservation of amino-acids.

The "correct" alignment for homologous sequences can not be computed. However, we can compute an optimal alignment.

This computation *(i)* assumes that pair-score based similarity measures are relevant, *(ii)* uses an empirical model of indel penalties, and ***(iii)*** **requires** $O(n^2)$ **computational resources.**

... and the $O(n^2)$ resource requirement means the algorithm is too slow for searches on a database scale i.e. in very large search spaces.



http://steipe.biochemistry.utoronto.ca/abc

B O R I S . S T E I P E @ U T O R O N T O . C A

DEPARTMENT OF BIOCHEMISTRY & DEPARTMENT OF MOLECULAR GENETICS
UNIVERSITY OF TORONTO, CANADA