CS612 - Algorithms in Bioinformatics

Sequence Alignment

February 10, 2025

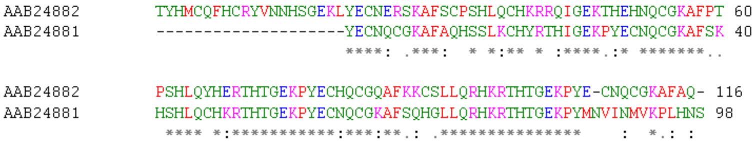
Searching for Sequence Similarity

- **Problem:** Determine possible biological function associated with a decoded gene sequence
- **Approach/Process:**
- Treat given gene sequence as a query sequence
- Search over a database of functionally-annotated gene sequences
    - Gene sequences for which the function is determined and deposited
- If the query sequence x is similar to a sequence y in the database
    - Then we add function(y) to the list of possible functions of x
- **Assumption:** similar sequences have similar functions
    - In other words, sequence is the main determinant of function

Searching for Sequence Similarity

- **Problem:** Determine possible biological function associated with a decoded gene sequence
- **Subproblems** (of general interest to computer scientists):
  - How do we measure sequence similarity?
  - How do we align two sequences? Do they have to match exactly or as long as they overlap significantly, we can make the same prediction?
  - Over what threshold of similarity does the assumption hold?
  - Can we associate a confidence as a function of similarity?
  - What if we want to compare more than two sequences?

Database Search and Sequence Alignment



```
AAB24882        TYHMCQFHCRYVNNHSGEKLYECNERSKAFSCPSHLQCHKRRQIGEKTHEHNQCGKAFPT 60
AAB24881        --------------------YECNQCGKAFAQHSSLKCHYRTHIGEKPYECNQCGKAFSK 40
                                    ****: .***:  * *:** * :****.:* *******..

AAB24882        PSHLQYHERTHTGEKPYECHQCGQAFKKCSLLQRHKRTHTGEKPYE-CNQCGKAFAQ- 116
AAB24881        HSHLQCHKRTHTGEKPYECNQCGKAFSQHGLLQRHKRTHTGEKPYMNVINMVKPLHNS 98
                 **** *:***********:***:**.: .***************    :  *.: :
```

Why do We Want to Compare Sequences?

- Evolutionary relationships
  - Phylogenetic trees can be constructed based on comparison of the sequences of a molecule (example: 16S rRNA) taken from different species
  - Residues conserved during evolution play an important role
- Prediction of protein structure and function
  - Proteins which are very similar in sequence generally have similar 3D structure and function as well
  - By searching a sequence of unknown structure against a database of known proteins the structure and/or function can in many cases be predicted

Database Search and Sequence Alignment

Definition (Sequence alignment/comparison)

The arrangement of two or more amino acid or nucleotide sequences in such a way as to maximize their similarity under some scoring function. Alternatively – we want to minimize the *edit distance* between the sequences

Definition (Edit distance)

The minimum number of **substitutions**, **deletions** or **insertions** required to convert one string into another

Database Search and Sequence Alignment

Example: How do we align ”kitten“ and ”sitting“?

1. **k**itten → **s**itten (substitution)
2. sitt**e**n → sitt**i**n (substitution)
3. sittin → sittin**g** (insertion)

```
KITTEN-
SITTING
```

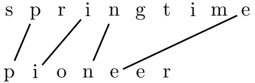
Longest Common Subsequence (LCS)

Definition

A *subsequence* of a sequence $A = \{a_1, a_2, \ldots, a_n\}$ is a sequence $B = \{b_1, b_2, \ldots, b_m\}$ (with $m \leq n$) such that

- Each b_i is an element of A.
- If b_i occurs before b_j in B (i.e., if $i < j$) then it also occurs before b_j in A.

- We do *not* assume that the elements of B are consecutive elements of A. For example: "axdy" is a subsequence of "baxefdoym"
- Given two sequences $X = \{x_1, x_2, \ldots, x_m\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$, the LCS is a subsequence common to both whose length is longest.



Things to Keep in Mind

- How do we determine the score?
  - What is the reward for a match? Same for all matches?
  - What it the penalty for a mismatch? Are all mismatches the same? (Usually not. We use substitution matrices to estimate this)
  - Gap penalty – Same penalty for opening a gap vs. extending it?
- How do we perform the alignment? (Dynamic programming or variants)
- How do we statistically evaluate the significance of our results?

Things to Keep in Mind When Working With Alignments

- Pairwise alignment programs always find the optimal alignment of two sequences
  - They do so even if it does not make any sense at all to align the two sequences
  - "Optimal" means optimal according to the **substitution matrix** and **gap penalties** you choose – also if you choose the wrong ones
- Generally the underlying assumptions are wrong
  - The frequency of substitution is not the same at all positions
  - Nor is the frequencies of insertions and deletions the same
  - Affine gap penalties do not properly model ins/del events

Using Sequence Alignment to Search Databases

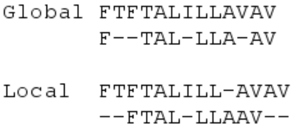
- The most common usage of pairwise sequence alignment is searching databases for related sequences
- Although the alignments themselves may be unreliable the alignment scores gives a lot of information about which sequences are related and which are not
- Having a set of related sequences is a lot more informative than just one sequence – even if nothing is known about the related sequences

Requirements for Sequence Alignment

- A very fast method to find potentially related sequences
  - Systematically searching through the databases with the alignment methods take too long even though dynamic programming is fast
  - Some method to initially identify possible matches is therefore needed to speed up the search
- A method to evaluate which matches to trust
  - Statistics on the alignment score distributions can be used to calculate the significance of an alignment
  - This way we can not only rank which matches are better than others but also tell if any of them are good at all

Local or Global Alignment

- Global alignment "forces" the alignment of the entire sequence.
- Generally local alignment is used for performing database searches
  - For most cases you would be interested in knowing if any parts of you sequences looks like something else
  - The protein sequence databases have not been split into domains
- It is not always the optimal thing to do but ...
  - In the case where the complete sequence should match the local alignment score will be almost identical to the global one
  - If you really want a global alignment you can make it afterwards



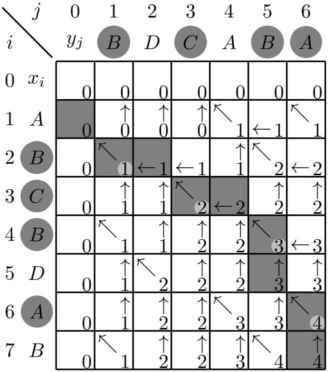
```
Global  FTFTALILLAVAV
        F--TAL-LLA-AV

Local   FTFTALILL-AVAV
        --FTAL-LLAAV--
```

Local or Global Alignment

- Because you can to start a new alignment anywhere dynamic programming scores cannot become negative
- The trace-back is started at the highest values rather than the lower right corner
- The trace-back is stopped as soon as a zero is encountered

Global Alignment – Generic Example

- Here we use the basic LCS for demonstration purposes.
- We allocate an $(m+1) \times (n+1)$ table, where m and n are the sizes of the sequences, plus a 0^{th} row and a 0^{th} column.
- The dynamic programming equation below tells us how to fill the table, from top to bottom and left to right.
- We add 1 for each match.
- In global alignment, $C[m,n]$ is the final result.



i \ j		0	1	2	3	4	5	6
		y_j	B	D	C	A	B	A
0	x_i	0	0	0	0	0	0	0
1	A	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	B	0	↖ 1	← 1	← 1	↑ 1	↖ 2	← 2
3	C	0	↑ 1	↑ 1	↖ 2	← 2	↑ 2	↑ 2
4	B	0	↖ 1	↑ 1	↑ 2	↑ 2	↖ 3	← 3
5	D	0	↑ 1	↖ 2	↑ 2	↑ 2	↑ 3	↑ 3
6	A	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	B	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

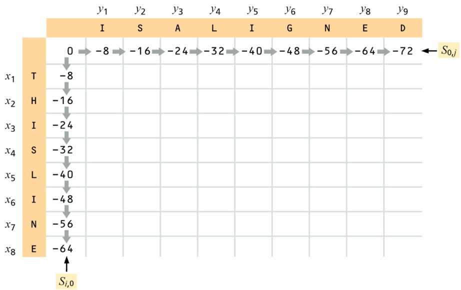
Global Alignment – Generic Example

We fill the table top to bottom, left to right, as:

$$c[i,j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ c[i-1,j-1]+1 & \text{if } i,j > 0 \text{ and } x_i = y_j \\ \max\{c[i-1,j], c[i,j-1]\} & \text{if } i,j > 0 \text{ and } x_i \neq y_j \end{cases}$$

- $c[i,j]$ represents the match score between $x[1\ldots i]$ and $y[1\ldots j]$.
- If any of the indices is 0, this is a match with an empty string, which is by definition 0.
- Our final score is $c[m,n]$.
- In sequence alignment we score matches/mismatches and gaps according to biological criteria.

Global Alignment: Needleman-Wunsch



			y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9
			I	S	A	L	I	G	N	E	D
		0	-8	-16	-24	-32	-40	-48	-56	-64	-72
x_1	T	-8									
x_2	H	-16									
x_3	I	-24									
x_4	S	-32									
x_5	L	-40									
x_6	I	-48									
x_7	N	-56									
x_8	E	-64									

$$S_{i,j} = \begin{cases} 0 & i,j = 0 \\ max(S_{i-1,j-1} + s(x_i, y_j), & \\ S_{i-1,j} + g, S_{i,j-1} + g) & otherwise \end{cases}$$

Optimal alignment:

```
THIS-LI-NE-
--ISALIGNED
```

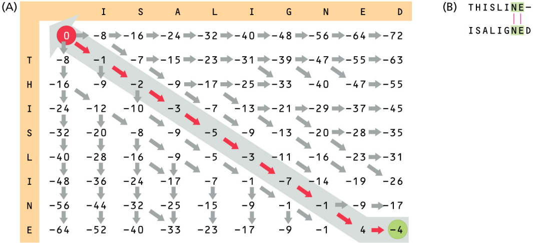
```
x = THISLINE
y = ISALIGNED
```

$S_{i,j}$: the score of the optimal alignment of all characters/amino acids up to x_i of x will all residues up to y_j of y.

The first row and columns are gaps.

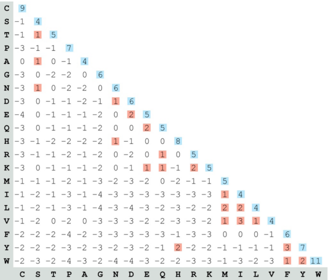
$s(x_i, y_i)$ is the mis/match score of x,y, and g is a gap penalty (-8 here).

Global Alignment: Needleman-Wunsch



		I	S	A	L	I	G	N	E	D
	0	-8	-16	-24	-32	-40	-48	-56	-64	-72
T	-8	-1	-7	-15	-23	-31	-39	-47	-55	-63
H	-16	-9	-2	-9	-17	-25	-33	-40	-47	-55
I	-24	-12	-10	-3	-7	-13	-21	-29	-37	-45
S	-32	-20	-8	-9	-5	-9	-13	-20	-28	-35
L	-40	-28	-16	-9	-5	-3	-11	-16	-23	-31
I	-48	-36	-24	-17	-7	-1	-7	-14	-19	-26
N	-56	-44	-32	-25	-15	-9	-1	-1	-9	-17
E	-64	-52	-40	-33	-23	-17	-9	-1	4	-4

```
THISLINE-
      ||
ISALIGNED
```



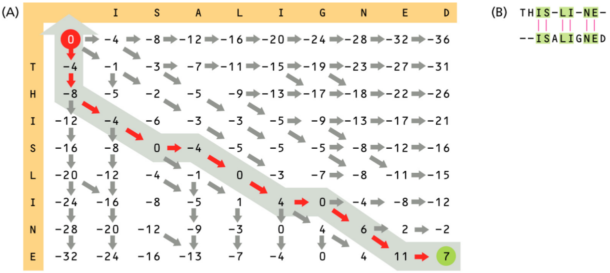
C	9																			
S	-1	4																		
T	-1	1	5																	
P	-3	-1	-1	7																
A	0	1	0	-1	4															
G	-3	0	-2	-2	0	6														
N	-3	1	0	-2	-2	0	6													
D	-3	0	-1	-1	-2	-1	1	6												
E	-4	0	-1	-1	-1	-2	0	2	5											
Q	-3	0	-1	-1	-1	-2	0	0	2	5									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

$$S_{i,j} = \begin{cases} 0 & i,j = 0 \\ max(S_{i-1,j-1} + s(x_i, y_j), \\ S_{i-1,j} + g, S_{i,j-1} + g) & otherwise \end{cases}$$

Scoring matrix used: BLOSUM-62

The gap penalty is so high (-8) that there is no incentive to add gaps rather than allow mismatches (the most severe of which has a penalty of -4) The "fault" is with the scoring matrix used the alignment is optimal within the scoring matrix used.

Global Alignment: Needleman-Wunsch



(A)

		I	S	A	L	I	G	N	E	D
	0	-4	-8	-12	-16	-20	-24	-28	-32	-36
T	-4	-1	-3	-7	-11	-15	-19	-23	-27	-31
H	-8	-5	-2	-5	-9	-13	-17	-18	-22	-26
I	-12	-4	-6	-3	-3	-5	-9	-13	-17	-21
S	-16	-8	0	-4	-5	-5	-5	-8	-12	-16
L	-20	-12	-4	-1	0	-3	-7	-8	-11	-15
I	-24	-16	-8	-5	1	4	0	-4	-8	-12
N	-28	-20	-12	-9	-3	0	4	6	2	-2
E	-32	-24	-16	-13	-7	-4	0	4	11	7

(B)

```
THIS-LI-NE-
  || || ||
--ISALIGNED
```

$$S_{i,j} = \begin{cases} 0 & i,j = 0 \\ max(S_{i-1,j-1} + s(x_i, y_j), & \\ S_{i-1,j} + g, S_{i,j-1} + g) & otherwise \end{cases}$$

This scoring matrix used matches the gap penalty (-4) to the most severe mismatch (-4).

From Global to Local Alignment

Main differences over Needleman-Wunsch:

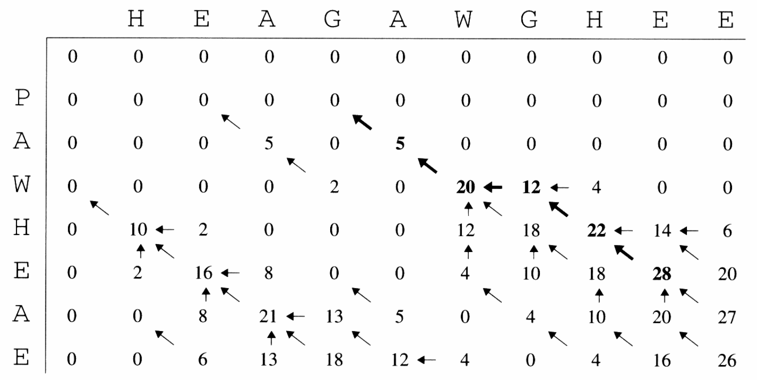
- Whenever the score of the optimal sub-alignment is less than zero, it is rejected (the matrix element is set to 0)
- Traceback starts from the highest-scoring element:

$$S_{i,j} = \max \begin{cases} S_{i-1,j-1} + s(x_i, y_j) \\ S_{i-1,j} + g(n_{gap1})_{1 \le n_{gap1} \le i} \\ S_{i,j-1} + g(n_{gap2})_{1 \le n_{gap2} \le j} \\ 0 \end{cases}$$

What does the rejection of a negative optimal sub-alignment mean?
Hint: many mini global alignments not worth to continue at some point

Note that the score given takes into account affine gap penalties (penalizing more for opening a gap, less for extending a gap)

The Smith-Waterman algorithm (local alignment)



		H	E	A	G	A	W	G	H	E	E
	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	5	0	**5**	0	0	0	0	0
W	0	0	0	0	2	0	**20**	**12**	4	0	0
H	0	10	2	0	0	0	12	18	**22**	14	6
E	0	2	16	8	0	0	4	10	18	**28**	20
A	0	0	8	21	13	5	0	4	10	20	27
E	0	0	6	13	18	12	4	0	4	16	26

```
AWGHE
AW-HE
```

An Example of a Substitution Matrix

	A	G	C	T
A	+1	-3	-3	-3
G	-3	+1	-3	-3
C	-3	-3	+1	-3
T	-3	-3	-3	+1

```
C A G G T A G C A A G C T T G C A T G T C A
| |   | | | | | | | | | | | |     | | | | |
C A C G T A G C A A G C T T G – G T G T C A
```

Score = 19-9 = 10

Types of Matrices

- **Percent Identity** – Standard scoring matrix to align DNA sequences
- **PAM** – Estimates the rate at which each possible residue in a sequence changes to each other residue over time
- **BLOSUM-X** – Identifies sequences that are X% similar to the query sequence

Nucleotide Scoring Matrix

Approximate ratios used on the web page:

Percent identity	Match/Mismatch
99%	1/-3
98%	2/-5
95%	1/-2
90%	2/-3
85%	3/-4
80%	4/-5
75%	1/-1
70%	11/-10
65%	5/-4
60%	7/-5
50%	3/-2

Amino Acid PAM Matrices

- **P**ercent **A**ccepted **M**utation
- Dayhoff (1978), 1572 changes in 71 families of proteins, at least 85% similar
- For each amino acid, count 20 numbers
- For example, how many F (phenylalanine) stay the same, how many change to the other 19 amino acids
- Normalize: divide each of these 20 numbers by (sum of 20 numbers)
- PAM1: 1% probability of change on average of all amino acid positions.

The Column/Row of F in PAM1

F to A: 0.0002
F to R: 0.0001
F to N: 0.0001
F to D: 0.0000
F to C: 0.0000
F to Q: 0.0000
F to E: 0.0000
F to G: 0.0001
F to H: 0.0002
F to I: 0.0007
F to L: 0.0013
F to K: 0.0000
F to M: 0.0001
F to F: 0.9946
F to P: 0.0001
F to S: 0.0003
F to T: 0.0001
F to W: 0.0001
F to Y: 0.0021
F to V: 0.0001

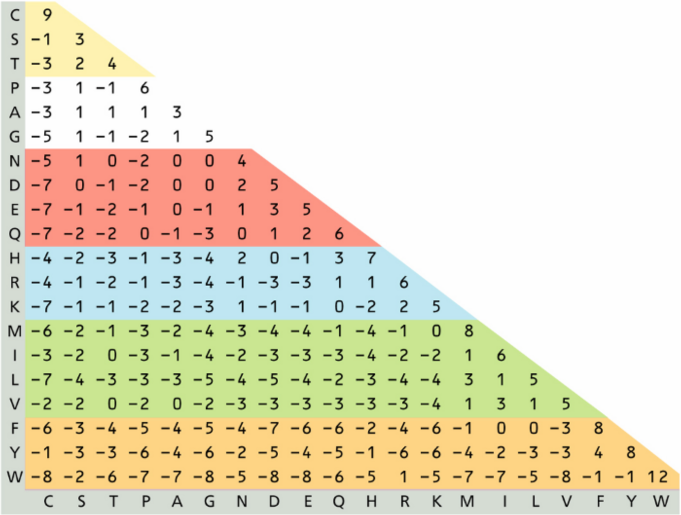
Compute PAM250

$$PAM_2 = PAM_1 * PAM_1 = (PAM_1)^2$$

$$PAM_{250} = (PAM_1)^{250}$$

- After 100 PAMs of evolution, not every residue will have changed: some will have mutated several times, perhaps returning to their original state, and others not at all.
- Therefore, it makes sense to use more than 100.
- It does not correspond to actual time, since different families evolve differently.

Example - PAM120



C	9																			
S	−1	3																		
T	−3	2	4																	
P	−3	1	−1	6																
A	−3	1	1	1	3															
G	−5	1	−1	−2	1	5														
N	−5	1	0	−2	0	0	4													
D	−7	0	−1	−2	0	0	2	5												
E	−7	−1	−2	−1	0	−1	1	3	5											
Q	−7	−2	−2	0	−1	−3	0	1	2	6										
H	−4	−2	−3	−1	−3	−4	2	0	−1	3	7									
R	−4	−1	−2	−1	−3	−4	−1	−3	−3	1	1	6								
K	−7	−1	−1	−2	−2	−3	1	−1	−1	0	−2	2	5							
M	−6	−2	−1	−3	−2	−4	−3	−4	−4	−1	−4	−1	0	8						
I	−3	−2	0	−3	−1	−4	−2	−3	−3	−3	−4	−2	−2	1	6					
L	−7	−4	−3	−3	−3	−5	−4	−5	−4	−2	−3	−4	−4	3	1	5				
V	−2	−2	0	−2	0	−2	−3	−3	−3	−3	−3	−3	−4	1	3	1	5			
F	−6	−3	−4	−5	−4	−5	−4	−7	−6	−6	−2	−4	−6	−1	0	0	−3	8		
Y	−1	−3	−3	−6	−4	−6	−2	−5	−4	−5	−1	−6	−6	−4	−2	−3	−3	4	8	
W	−8	−2	−6	−7	−7	−8	−5	−8	−8	−6	−5	1	−5	−7	−7	−5	−8	−1	−1	12
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W

BLOSUM Matrices

- **BLO**cks of amino acid **SU**bstitution **M**atrices
- Start with highly-conserved patterns (blocks) in a large set of closely related proteins
- Use the likelihood of substitutions found in those sequences to create a substitution probability matrix
- BLOSUM-n means that the sequences used were n% alike
- BLOSUM62 is “standard”
- Nature Biotechnology: `http://www.nature.com/nbt/journal/v22/n8/abs/nbt0804-1035.html`

Example of BLOSUM62

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

Negative for less likely substitutions

Positive for more likely substitutions

Common amino acids have low weight

Rare amino acids have high weight

Which Scoring Matrix to Use?

- How can one decide whether to use BLOSUM or PAM when comparing and aligning sequences?
- This decision is also more difficult when the evolutionary distance between the sequences is not known
- What to do: try different ones and compare results
- Different studies have concluded that for the PAM matrices it is generally best to try PAM40, PAM120, and PAM250
- When used for local alignments
    - Lower PAM matrices find short local alignments
    - Higher PAM matrices find longer but weaker local alignments
- Several different matrices should be used, and the alignment that is judged to be evolutionarily the most accurate is the one chosen
    - Question: how can one judge which one is the most accurate?
    - Judgment on a control set where the evolutionary relationship is known

Heuristic Search Algorithms

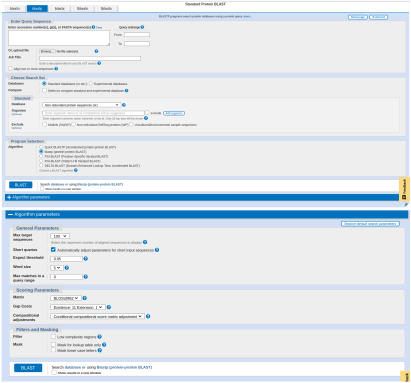
- FASTA (Pearson 1995)
- Uses heuristics to avoid calculating the full dynamic programming matrix
- Speed up searches by an order of magnitude compared to full Smith-Waterman
- The statistical side of FASTA is still stronger than BLAST

- BLAST (Altschul 1990, 1997)
- Uses rapid word lookup methods to completely skip most of the database entries
- Extremely fast
- Almost as sensitive as FASTA

BLAST at NCBI

`http://www.ncbi.nlm.nih.gov/BLAST/`

- Very fast computer dedicated to running BLAST searches
- Many databases that are always up to date
- Nice simple web interface
- But you still need to knowledge about BLAST to use it properly



Enter Query Sequence

Enter accession number(s), gi(s), or FASTA sequence(s) Clear

Query subrange: From □ To □

Or, upload file: Browse... No file selected.

Job Title: Enter a descriptive title for your BLAST search

□ Align two or more sequences

Choose Search Set

Databases: ● Standard databases (nr etc.): ○ Experimental databases

Compare: □ Select to compare standard and experimental database

Standard

Database: Non-redundant protein sequences (nr)

Organism (Optional): Enter organism name or id--completions will be suggested □ exclude Add organism

Enter organism common name, binomial, or tax id. Only 20 top taxa will be shown

Exclude (Optional): □ Models (XM/XP) □ Non-redundant RefSeq proteins (WP) □ Uncultured/environmental sample sequences

Program Selection

Algorithm:
○ Quick BLASTP (Accelerated protein-protein BLAST)
● blastp (protein-protein BLAST)
○ PSI-BLAST (Position-Specific Iterated BLAST)
○ PHI-BLAST (Pattern Hit Initiated BLAST)
○ DELTA-BLAST (Domain Enhanced Lookup Time Accelerated BLAST)

Choose a BLAST algorithm

BLAST — Search database nr using Blastp (protein-protein BLAST)

□ Show results in a new window

+ Algorithm parameters

— Algorithm parameters

General Parameters

Max target sequences: 100
Select the maximum number of aligned sequences to display

Short queries: ☑ Automatically adjust parameters for short input sequences

Expect threshold: 0.05

Word size: 5

Max matches in a query range: 0

Scoring Parameters

Matrix: BLOSUM62

Gap Costs: Existence: 11 Extension: 1

Compositional adjustments: Conditional compositional score matrix adjustment

Filters and Masking

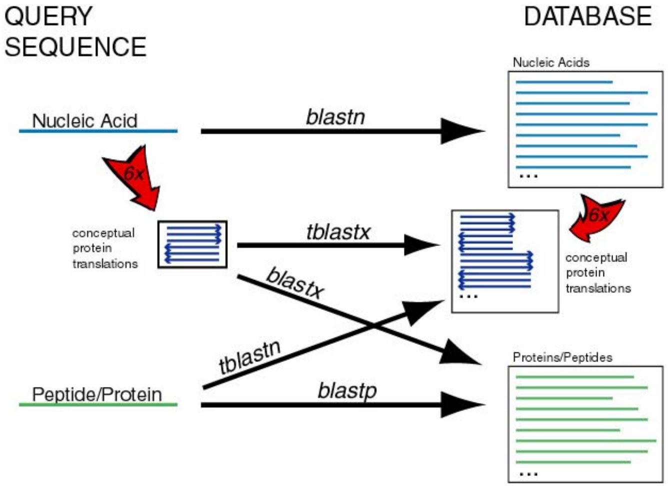
Filter: □ Low complexity regions

Mask: □ Mask for lookup table only
□ Mask lower case letters

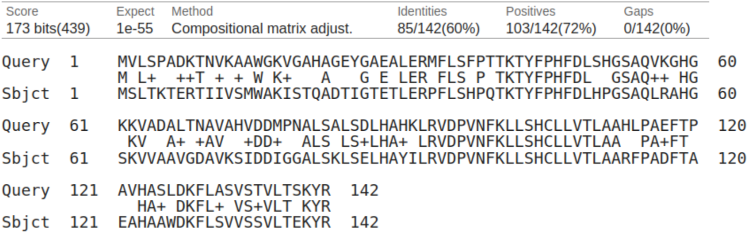
BLAST — Search database nr using Blastp (protein-protein BLAST)

□ Show results in a new window

Different BLAST Programs



Pairwise alignment of hemoglobin α chain and ζ chain



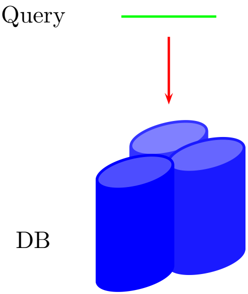
Score	Expect	Method	Identities	Positives	Gaps
173 bits(439)	1e-55	Compositional matrix adjust.	85/142(60%)	103/142(72%)	0/142(0%)

```
Query  1    MVLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHFDLSHGSAQVKGHG  60
            M L+  ++T + + W K+   A   G E LER FLS P TKTYFPHFDL  GSAQ++ HG
Sbjct  1    MSLTKTERTIIVSMWAKISTQADTIGTETLERPFLSHPQTKTYFPHFDLHPGSAQLRAHG  60

Query  61   KKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTP  120
             KV  A+ +AV  +DD+  ALS LS+LHA+ LRVDPVNFKLLSHCLLVTLAA  PA+FT
Sbjct  61   SKVVAAVGDAVKSIDDIGGALSKLSELHAYILRVDPVNFKLLSHCLLVTLAARFPADFTA  120

Query  121  AVHASLDKFLASVSTVLTSKYR  142
              HA+ DKFL+ VS+VLT KYR
Sbjct  121  EAHAAWDKFLSVVSSVLTEKYR  142
```

How BLAST Works

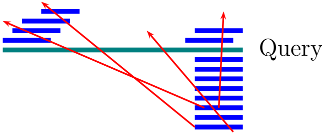
Basic **L**ocal **A**lignment **S**earch **T**ool
Main idea:

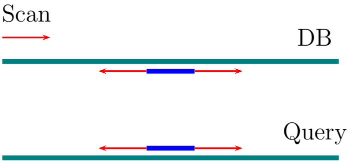
- Construct a dictionary of all the words in the query
- Initiate a local alignment for each word match between query and DB
- Running Time: O(MN)
- However, orders of magnitude faster than Smith-Waterman



Blast – Original Version

- **Dictionary:** All words of length k (approx. 11) Alignment initiated between words of alignment score approx. T (typically T = k)
- **Alignment:** Ungapped extensions until score below statistical threshold
- **Output:** All local alignments with score more than statistical threshold





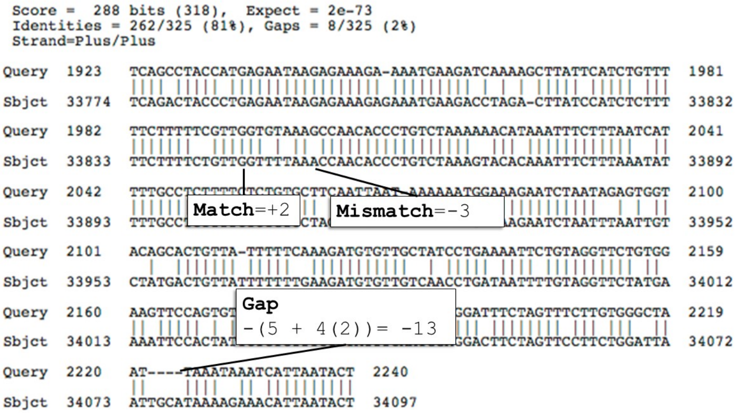
How BLAST works

- The search is accelerated by indexing the sequence databases in a so-called suffix array
  - Three letter subsequences are used as keys to the sequences
  - Closely related substitutions are also included
  - This gives approx. 150 index keys for each sequence
- This is used in two ways
  - To quickly discard sequences that are not similar at all before even beginning to align them
  - To constrain the alignment and thereby speed up the alignment procedure itself

Evaluating the Significance of an Alignment

- **Raw score:** depend on scoring matrix and method.
- **Identities** How many positions are identical.
- **Positives:** How many positions yield a positive score.
- **Gaps:** How many gaps there are.
- **E-value (Expect value):** number of unrelated database sequences expected to yield same or higher score by pure chance.

Nucleotide Scoring Example



```
 Score =  288 bits (318),  Expect = 2e-73
 Identities = 262/325 (81%), Gaps = 8/325 (2%)
 Strand=Plus/Plus

Query  1923   TCAGCCTACCATGAGAATAAGAGAAAGA-AAATGAAGATCAAAAGCTTATTCATCTGTTT  1981
              |||| ||||| ||||||||||||||||| ||||||||| | | | ||||| ||||| |||
Sbjct  33774  TCAGACTACCCTGAGAATAAGAGAAAGAGAAATGAAGACCTAGA-CTTATCCATCTCTTT  33832

Query  1982   TTCTTTTTCGTTGGTGTAAAGCCAACACCCTGTCTAAAAAACATAAATTTCTTTAATCAT  2041
              |||||||  |||||| | || |||||||||||||||||  ||| ||||||||||||  ||
Sbjct  33833  TTCTTTTCTGTTGGTTTTAAACCAACACCCTGTCTAAAGTACACAAATTTCTTTAAATAT  33892

Query  2042   TTTGCCTCTTTTCTCTGTGCTTCAATTAAT-AAAAAATGGAAAGAATCTAATAGAGTGGT  2100
              ||||||             ||                     ||||||||||  | | ||
Sbjct  33893  TTTGCCT            CTA                    AGAATCTAATTTAATTGT  33952

Query  2101   ACAGCACTGTTA-TTTTTCAAAGATGTGTTGCTATCCTGAAAATTCTGTAGGTTCTGTGG  2159
                |  ||||||| |||||  |||||||||||  | ||||| |||| |||||||||| ||
Sbjct  33953  CTATGACTGTTATTTTTTTGAAGATGTGTTGTCAACCTGATAATTTTGTAGGTTCTATGA  34012

Query  2160   AAGTTCCAGTGT                        GGATTTCTAGTTTCTTGTGGGCTA  2219
              || ||||| | |                        ||| |||||||| ||| |||  ||
Sbjct  34013  AAATTCCACTAT                        GGACTTCTAGTTCCTTCTGGATTA  34072

Query  2220   AT----TAAATAAATCATTAATACT  2240
              ||    ||||  || |||||||||| 
Sbjct  34073  ATTGCATAAAAGAAACATTAATACT  34097
```

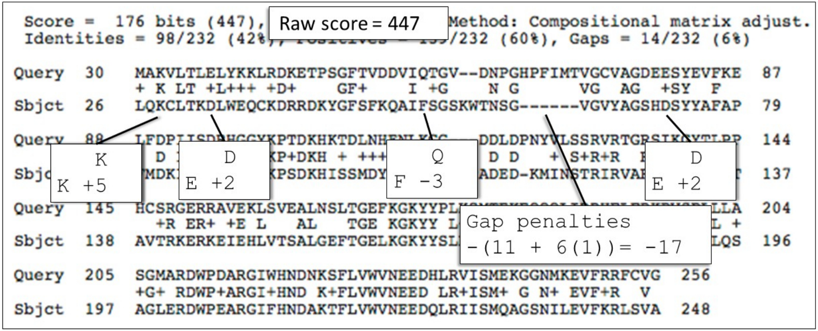
Match=+2

Mismatch=-3

Gap

-(5 + 4(2))= -13

Nucleotide Scoring Example



```
 Score =  176 bits (447),                    Method: Compositional matrix adjust.
 Identities = 98/232 (42%),                /232 (60%), Gaps = 14/232 (6%)

Query  30    MAKVLTLELYKKLRDKETPSGFTVDDVIQTGV--DNPGHPFIMTVGCVAGDEESYEVFKE  87
             + K LT +L+++ +D+     GF+     I +G    N G        VG  AG  +SY   F
Sbjct  26    LQKCLTKDLWEQCKDRRDKYGFSFKQAIFSGSKWTNSG------VGVYAGSHDSYYAFAP  79

Query  88    LFDP                             DDLDPNYVLSSRVRTGR           144
               D          KP+DKH + +++            D D    + S+R+R
Sbj           MDK         KPSDKHISSMDY         ADED-KMINSTRIRVA         137

Query  145   HCSRGERRAVEKLSVEALNSLTGEFKGKYYPL                        LLA  204
              +R ER+ +E L    AL    TGE KGKYY L                         L +
Sbjct  138   AVTRKERKEIEHLVTSALGEFTGELKGKYYSL                        LQS  196

Query  205   SGMARDWPDARGIWHNDNKSFLVWVNEEDHLRVISMEKGGNMKEVFRRFCVG  256
             +G+ RDWP+ARGI+HND K+FLVWVNEED LR+ISM+ G N+ EVF+R  V
Sbjct  197   AGLERDWPEARGIFHNDAKTFLVWVNEEDQLRIISMQAGSNILEVFKRLSVA  248
```

Raw score = 447

K
K +5

D
E +2

Q
F -3

D
E +2

Gap penalties

-(11 + 6(1))= -17

Evaluating the Significance of an Alignment

- The E-value describes the number of hits one can "expect" to see by chance when searching a database of a particular size.
- It decreases exponentially with the Score (S) that is assigned to a match between two sequences.
- It essentially describes the random background noise that exists for matches between sequences.
- The E-value is used as a convenient way to create a significance threshold for reporting results.
- When increased from the default value of 10, a larger list with more low-scoring hits can be reported.
- E-value approaching zero → significant alignment. Less than 0.01 = almost always homologous; 1e-10 for nucleotide searches of 1e-4 for protein searches= frequently related

Notice

- In BLAST 2.0, the E-value is also used instead of the P-value (probability) to report the significance of matches. For example, an E value of 1 assigned to a hit can be interpreted as meaning that in a database of the current size one might expect to see 1 match with a similar score simply by chance.
- Be careful when comparing E-values from different searches.
- Comparison is only meaningful for different query sequences searched against the same database with the same BLAST parameters.

Why Use Substitution Matrices?

- Determine likelihood of homology between two sequences.
- Substitutions that are more likely should get a higher score,
- Substitutions that are less likely should get a lower score.
- Segment pairs whose scores can not be improved by extension or trimming.
- These are called high-scoring segment pairs or HSPs.

Scoring Matrices

- Log-odds matrix where each cell gives the probability of aligning those two residues
- To analyze how high a score is likely to arise by chance, a model of random sequences is needed
- For proteins, the simplest model chooses the amino acid residues in a sequence independently, with specific background probabilities for the various residues.
- The expected score for aligning a random pair of amino acid should be negative (or long sequences will always get high scores)
- Score of alignment = Sum of log-odds scores of residues

Scoring Matrices

- In the limit of sufficiently large sequence lengths m and n, the statistics of High Scoring Segment Pairs (HSP) scores are characterized by two parameters, K and λ.
- The expected number of HSPs with score at least S is given by the formula:
- The E-value, the expected number of HSPs of lengths m and n with score at least S is given by:

$$E = Kmne^{-\lambda S}$$

- This formula makes eminently intuitive sense: Doubling the length of either sequence should double the number of HSPs attaining a given score.
- The value also decreases exponentially with the score.
- K and λ can be thought of simply as natural scales for the search space size and the scoring system respectively.

Scoring Matrices – Bit Scores

- The raw score has to be normalized. Define the bit score S' as follows:

$$S' = \frac{\lambda S - \ln K}{\ln 2}$$

- Then, by applying some log rules, we get:

$$E = mn2^{-S'}$$

How to Interpret Log-odds Matrix

- If you know the scores in a matrix, how do determine what kind of alignments it will find?
- You need to determine the frequencies implied by the scores

$$s(a,b) = \frac{1}{\lambda}\ln\left(\frac{p_{ab}}{f_a f_b}\right) \Rightarrow f_a f_b e^{\lambda s(a,b)} = p_{ab}$$

- where the p_{ab}, called target frequencies, are positive numbers that sum to 1 (representing all target substitutions), the f_i are background frequencies of amino acids, and λ is a positive constant, same as above.
- Work backwards to find the substitution frequencies.

How to Interpret Log-odds Matrix

In order to find p_{ab}, you need to find λ.

$$f_a f_b e^{\lambda s(a,b)} = p_{ab}$$

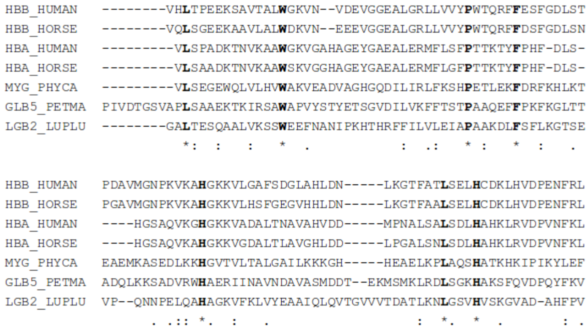
All probabilities must add up to 1, to set it to 1 and solve for lambda

$$\sum_{a,b} f_a f_b e^{\lambda s(a,b)} = 1$$

A Curse or a Blessing?

- Large databases are a blessing ...
  - They are more likely to contain something similar to the query
- ... and a curse
  - Increasing the size of the database decreases the significance of the hits you get
  - Searching huge databases requires fast computer
- What requirements this puts on software development
  - The programs must be speeded up or database searches will take longer and longer
  - The false positive rate must be reduced to not lose specificity

Multiple Sequence Alignment (MSA)

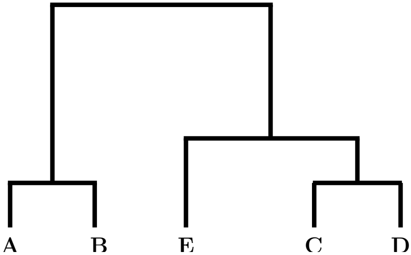


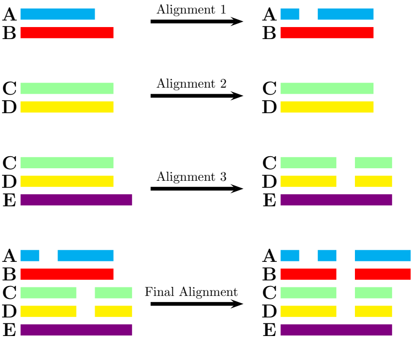
```
HBB_HUMAN    --------VHLTPEEKSAVTALWGKVN--VDEVGGEALGRLLVVYPWTQRFFESFGDLST
HBB_HORSE    --------VQLSGEEKAAVLALWDKVN--EEEVGGEALGRLLVVYPWTQRFFDSFGDLSN
HBA_HUMAN    ---------VLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-DLS-
HBA_HORSE    ---------VLSAADKTNVKAAWSKVGGHAGEYGAEALERMFLGFPTTKTYFPHF-DLS-
MYG_PHYCA    ---------VLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKSHPETLEKFDRFKHLKT
GLB5_PETMA   PIVDTGSVAPLSAAEKTKIRSAWAPVYSTYETSGVDILVKFFTSTPAAQEFFPKFKGLTT
LGB2_LUPLU   --------GALTESQAALVKSSWEEFNANIPKHTHRFFILVLEIAPAAKDLFSFLKGTSE
                       *:  :   :   *  .           :  .:   * :   *  :   .

HBB_HUMAN    PDAVMGNPKVKAHGKKVLGAFSDGLAHLDN-----LKGTFATLSELHCDKLHVDPENFRL
HBB_HORSE    PGAVMGNPKVKAHGKKVLHSFGEGVHHLDN-----LKGTFAALSELHCDKLHVDPENFRL
HBA_HUMAN    ----HGSAQVKGHGKKVADALTNAVAHVDD-----MPNALSALSDLHAHKLRVDPVNFKL
HBA_HORSE    ----HGSAQVKAHGKKVGDALTLAVGHLDD-----LPGALSNLSDLHAHKLRVDPVNFKL
MYG_PHYCA    EAEMKASEDLKKHGVTVLTALGAILKKKGH-----HEAELKPLAQSHATKHKIPIKYLEF
GLB5_PETMA   ADQLKKSADVRWHAERIINAVNDAVASMDDT--EKMSMKLRDLSGKHAKSFQVDPQYFKV
LGB2_LUPLU   VP--QNNPELQAHAGKVFKLVYEAAIQLQVTGVVVTDATLKNLGSVHVSKGVAD-AHFPV
                   . .:: *.  :   .                  :  *.  *  .       : .
```

Why MSA is Better?

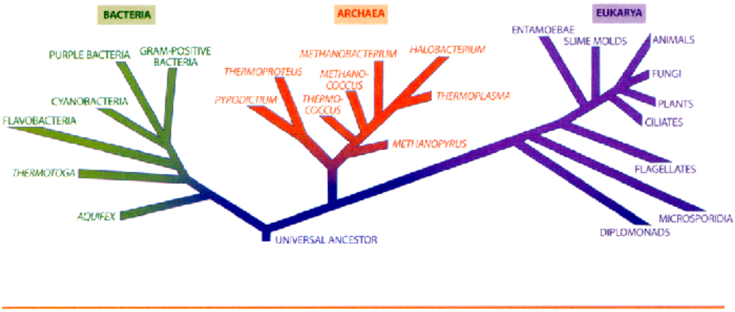
- More sequences contain more information
- Multiple sequence alignment allows us to compare all related proteins simultaneously
- It allows us to identify features that are conserved among the sequences
- Using a multiple sequence alignment (a profile) one can find more related sequences than by simple pairwise comparison

Building a Phylogenetic Tree





Assembling the Tree of Life



M. Madigan and B. Marrs, 1997

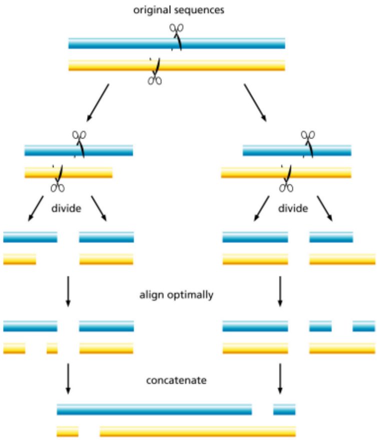
Assembled from aligned sequences of ribosomal RNA

Multiple Sequence Alignment

- Multiple sequence alignment is NP-hard.
- The most practical and widely used method in multiple sequence alignment is the hierarchical extensions of pairwise alignment methods.
- The principal is that multiple alignments is achieved by successive application of pairwise methods.

Divide and Conquer

- Divide the sequences near their midpoint.
- Repeat until length falls below threshold.
- Feed sequences to MSA.
- Merge sequences.



Multiple Sequence Alignment – Summary of Steps

- Compare all sequences pairwise.
- Perform cluster analysis on the pairwise data to generate a hierarchy for alignment. This may be in the form of a binary tree (guide tree).
- Build the multiple alignment by first aligning the most similar pair of sequences, then the next most similar pair and so on.
- Once an alignment of two sequences has been made, then this is fixed.
- Thus for a set of sequences A, B, C, D having aligned A with C and B with D the alignment of A, B, C, D is obtained by comparing the alignments of A and C with that of B and D using averaged scores at each aligned position.

	v_1	v_2	v_3	v_4
v_1	–			
v_2	.17	–		
v_3	.87	.28	–	
v_4	.59	.33	.62	–

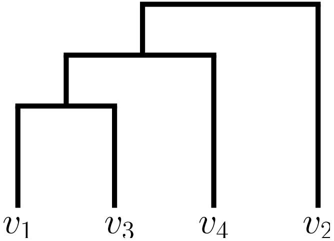
.17 means 17% identical.

Calculate:

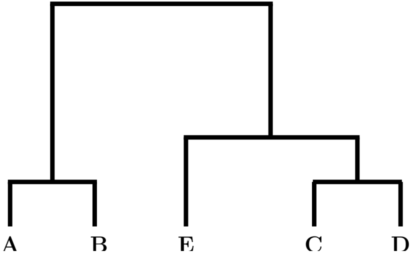
$v_{1,3} = alignment(v_1, v_3)$

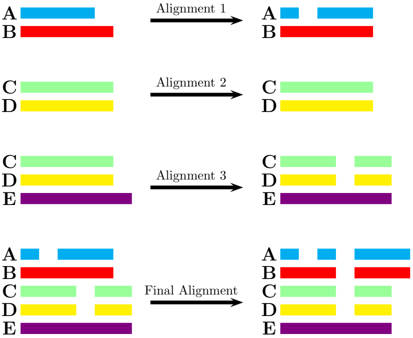
$v_{1,3,4} = alignment((v_{1,3}), v_4)$

$v_{1,2,3,4} = alignment((v_{1,3,4}), v_2)$



Example


A
B
E
C
D


A
B
Alignment 1
A
B
C
D
Alignment 2
C
D
C
D
E
Alignment 3
C
D
E
A
B
C
D
E
Final Alignment
A
B
C
D
E

Building a Consensus Sequence

- Concatenation of all the sequences can give a consensus sequence
- The consensus character for column i is the character that minimizes the summed distance to it from all the characters in column i
- Distance is measured using the substitution matrix
- A very simple method, but doesn't account for variability.
- Useful for highly conserved sequences.

A	B	A
A	B	–
–	B	A
C	A	–
A	B	A

Sequence Patterns

Patterns are known as regular expressions.

- The Prosite syntax for patterns:
  - Uses one-letter codes for amino acids (G=Gly, P=Pro, ...)
  - Each element in a pattern is separated from its neighbor by a $'-'$
  - The symbol $'X'$ is used where any amino acid is accepted
  - Ambiguities are indicated by square parentheses $'[]'$ ([AG] means Ala or Gly)
  - Amino acids that are not accepted at a given position are listed between a pair of curly brackets $'\{\}'$ ({AG} means any amino acid except Ala and Gly),
  - Repetitions are indicated between parentheses $'()'$ ([AG](2,4) means Ala or Gly between 2 and 4 times, X(2) means any amino acid twice).
  - A pattern is anchored to the first and last positions in the protein by the symbols $' <'$ and $' >'$ respectively.

Sequence Patterns – Example

The following pattern: $< A - x - [ST](2) - x(0,1) - \{V\}$ means:

- An Alanine (A) in the first position
- Followed by any amino acid,
- Followed by a Serine (S) or Threonine (T) twice.
- Followed or not by any amino acid.
- Followed by any amino acid except Valine (V).

How to Build a Pattern

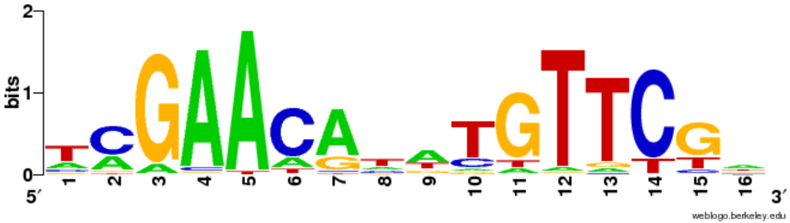
```
GHEGVGKVVKLGAGA
GHEKKGYFEDRGPSA
GHEGYGGRSRGGGYS
GHEFEGPKGCGALYI
GHELRGTTFMPALEC
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
G	H	E	G	V	G	K	V	V	K	L	G	A	G	A
			K	K		Y	F	E	D	R	A	P	S	S
			F	Y		G	R	S	R	G		G	Y	I
			L	E		P	K	G	C	P		L	E	C
				R		T	T	F	M					

Pattern: $G - H - E - X(2) - G - X(5) - [GA] - X(3)$

Search databases

Sequence Logo



Pros and Cons of Profiles

- Fast and easy to implement and understand.
- Unlike a consensus sequence – can accommodate alternative amino acids per position.
- Not sensitive to insertions/deletions.
- Small patterns find a lot of false positives. Long patterns are very difficult to design.

Searching Similar Sequences Using PSI-BLAST

PSI-Blast = Position Specific Iterated BLAST.

- A standard BLAST search is performed against a database using a substitution matrix (e.g. BLOSUM62).
- A position-specific scoring matrix (PSSM) is constructed automatically from a multiple alignment of the highest scoring hits of the initial BLAST search. High conserved positions receive high scores and weakly conserved positions receive low scores.
- The PSSM replaces the initial matrix to perform a second BLAST search.
- The former steps can be repeated and the new found sequences included to build a new PSSM.
- We say that the PSI-BLAST has converged if no new sequences are included in the last cycle.

PSI-BLAST dangers

- Avoid too close sequences → overfit!
- Can include false homologous! Therefore check the matches carefully: include or exclude sequences based on biological knowledge.
- The E-value reflects the significance of the match to the previous training set, not to the original sequence!
- Choose carefully your query sequence.
- Try reverse experiment to certify.

Clustal Omega for Multiple Sequence Alignment

- Clustal Omega can create multiple alignments, manipulate existing alignments, do profile analysis and create phylogentic trees.
- Scoring alignments by calculating all the pairwise scores and progressively build a tree using a neighbor joining algorithm.
- Uses HMM and several clustering methods.

Other State-of-the-art Methods

- MUSCLE – MUltiple Sequence Comparison by Log-Expectation. Significantly faster than ClustalW and often gives better results.
- T-Coffee (Tree-based Consistency Objective Function For alignment Evaluation).
- MAFFT (Multiple Alignment using Fast Fourier Transform).
- For a full list see here: `https://www.ebi.ac.uk/jdispatcher/msa`

General Considerations for MSA

- The more sequences to align the better.
- Don't include similar ($> 80\%$) sequences.
- Sub-groups should be pre-aligned separately, and one member of each subgroup should be included in the final multiple alignment.

HMMer - Turn an Alignment into a Sequence Profiles

Start with a multiple sequence alignment

↓

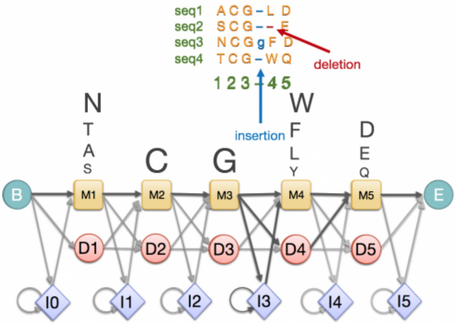
Insertions / deletions can be modelled

↓

Occupancy and amino acid frequency at each position in the alignment are encoded

↓

Profile created



Sources Cited

- Debra Goldberg, Algorithms for Molecular Biology, Fall 2008 `www.bioalgorithms.info` (lectures for students and faculty).
- Daniel Sam, "Greedy Algorithm" presentation.
- Glenn Tesler, "Genome Rearrangements in Mammalian Evolution: Lessons from Human and Mouse Genomes" presentation.
- Ernst Mayr, "What evolution is".
- Neil C. Jones, Pavel A. Pevzner, "An Introduction to Bioinformatics Algorithms" .
- Alberts, Bruce, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, Peter Walter. Molecular Biology of the Cell. New York: Garland Science. 2002.
- Mount, Ellis, Barbara A. List. Milestones in Science & Technology. Phoenix: The Oryx Press. 1994.
- Voet, Donald, Judith Voet, Charlotte Pratt. Fundamentals of Biochemistry. New Jersey: John Wiley & Sons, Inc. 2002.
- Campbell, Neil. Biology, Third Edition. The Benjamin/Cummings Publishing Company, Inc., 1993.
- Snustad, Peter and Simmons, Michael. Principles of Genetics. John Wiley & Sons, Inc, 2003.
- The BLAST manual.