**Sequence Analysis and Structure Prediction Service**

Centro Nacional de Biotecnología – CSIC

8-10 May, 2013

# Introductory course on Multiple Sequence Alignment

# Part I: Theoretical foundations

Course Notes

Instructor:

Mónica Chagoyen

monica.chagoyen@cnb.csic.es

# Contents

*Contents* ........ *3*

*Introduction* ........ *5*

**Some notes on protein evolution** ........ **6**
Mutations ........ 7
Variation among species ........ 7
Variation within species ........ 7
Domain shuffling ........ 8

*Finding sequences to align* ........ *9*

**Fundamentals: pair-wise sequence alignment** ........ **9**
Similarity scores ........ 9
Algorithms (Dynamic programming) ........ 12

**Basic Local Alignment Search Tool (BLAST)** ........ **13**
The algorithm ........ 13
Significance of an alignment score ........ 14

*Performing multiple sequence alignment* ........ *15*

**Progressive methods** ........ **16**
ClustalW ........ 16
T-Coffee ........ 16
MUSCLE ........ 17

**Iterative methods** ........ **17**
MAFFT ........ 17

**Choosing the right MSA software** ........ **18**

**Alignment visualization and editing** ........ **18**
Jalview ........ 19

**Representing multiple sequence alignments** ........ **20**
Consensus sequence ........ 20
Sequence logos ........ 20

*Working with profiles* ........ *23*

**Sequence patterns** ........ **23**
PROSITE patterns ........ 24

**Position-specific scoring matrices (PSSM)** ........ **24**

**Hidden Markov Models (HMM)** ........ **25**

**Domain/Family databases** ........ **25**
Pfam ........ 26
SMART ........ 26
InterPro ........ 26

**Profile searches** ........ **27**
Sequence-profile searches ........ 27

Profile-sequence searches ........ 27
Profile-profile methods ........ 28

***Content sources and bibliography ........ 29***

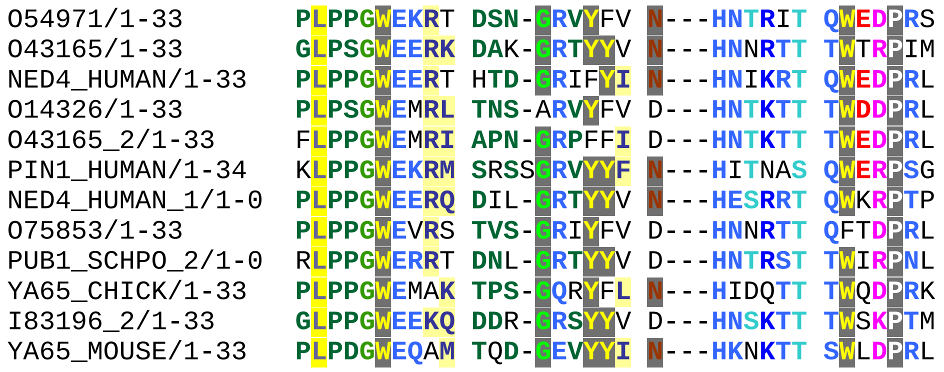
**Online resources ........ 29**

**Books and articles ........ 29**

# Introduction

A multiple sequence alignment (MSA) is a sequence alignment of three or more biological sequences, generally protein or DNA. In many cases, the input set of query sequences are assumed to have an evolutionary relationship by which they share a lineage and are descended from a common ancestor.

Example: A multiple sequence alignment corresponding to the WW domain (Source: SMART database)



```
O54971/1-33        PLPPGWEKRT DSN-GRVYFV N---HNTRIT QWEDPRS
O43165/1-33        GLPSGWEERK DAK-GRTYYV N---HNNRTT TWTRPIM
NED4_HUMAN/1-33    PLPPGWEERT HTD-GRIFYI N---HNIKRT QWEDPRL
O14326/1-33        PLPSGWEMRL TNS-ARVYFV D---HNTKTT TWDDPRL
O43165_2/1-33      FLPPGWEMRI APN-GRPFFI D---HNTKTT TWEDPRL
PIN1_HUMAN/1-34    KLPPGWEKRM SRSSGRVYYF N---HITNAS QWERPSG
NED4_HUMAN_1/1-0   PLPPGWEERQ DIL-GRTYYV N---HESRRT QWKRPTP
O75853/1-33        PLPPGWEVRS TVS-GRIYFV D---HNNRTT QFTDPRL
PUB1_SCHPO_2/1-0   RLPPGWERRT DNL-GRTYYV D---HNTRST TWIRPNL
YA65_CHICK/1-33    PLPPGWEMAK TPS-GQRYFL N---HIDQTT TWQDPRK
I83196_2/1-33      GLPPGWEEKQ DDR-GRSYYV D---HNSKTT TWSKPTM
YA65_MOUSE/1-33    PLPDGWEQAM TQD-GEVYYI N---HKNKTT SWLDPRL
```

Multiple sequence alignment also refers to the process of aligning such sequence set. Because three or more sequences of biologically relevant length can be difficult and are almost always time-consuming to align by hand, computational algorithms are used to produce and analyze the alignments. MSAs require more sophisticated methodologies than pair-wise alignments because they are more computationally complex. Most multiple alignment programs use heuristic methods (see box) rather than global optimization because identifying the optimal alignment between more than a few sequences of moderate length is prohibitively computationally expensive.

Heuristic: In computer science, a heuristic is a technique designed to solve a problem that ignores whether the solution can be proven to be correct, but which usually produces a good solution or solves a simpler problem that contains or intersects with the solution of the more complex problem.

Alignment is not a single problem but rather a collection of many quite diverse questions that all have in common the search for sequence similarity. Starting from the definition of alignment, there are two biologically meaningful formulations – one based on the desire to find evolutionary relationships and one based on the desire to find putative functional relationships.

Given the amino acid sequences of a set of proteins to be compared, an alignment displays the residues for each protein on a single line, with gaps ("–") inserted such that "equivalent" residues appear in the same column. The precise meaning of equivalence is generally context dependent: for the phylogeneticist, equivalent residues have common evolutionary ancestry; for the structural biologist, equivalent residues correspond to analogous positions belonging to homologous folds in a set of proteins; for the molecular biologist, equivalent residues play similar functional roles in their corresponding proteins. In each case, an alignment provides a bird's eye view of the underlying evolutionary, structural, or functional constraints characterizing a protein family in a concise, visually intuitive format.

Many bioinformatic methods rely on MSAs. Success of methods relies on quality of alignment. MSA a difficult problem, although current automatic approaches are good, usually they can be improved by human intervention to yield better alignments and hence better analysis. It is difficult to teach how to do this well - depends on obtaining experience. This course aims to provide a starting point to obtain this experience.

A typical multiple sequence alignment workflow will be:

1. Clearly formulate the question you want to answer. E.g.: "What is the secondary structure prediction for protein sequence"

2. Collect a set of sequences to address this question. E.g. using a BLAST database search.

3. Create and examine initial automatic MSA. E.g. align using Clustal and examine in JalView

4. Adjust the set of sequences to obtain an optimal set. E.g. remove unrelated sequences, add sequences from additional searches.

5. Manually adjust alignment to correct automatically-introduced errors.

## Some notes on protein evolution

The evolutionary variations of a protein provide much information about the protein itself. **Evolutionary divergence** into different species has resulted in many variants of the same protein, all with essentially the same biological function but different amino acid sequences. The differences and similarities of the amino acid sequences of these variants reflect the constraints of structure and function for that protein.

The number of possible RNA, DNA, or protein sequences is so great that it is implausible that similar long sequences could have arisen by any mechanism other than evolutionary divergence from the same ancestor.

Nucleic acids and proteins that have evolved from a common ancestor are said to be **homologous**. The sequences of homologous genes and proteins were identical at the time they originated by replication of a single gene. Subsequently the two genes accumulate mutational changes, and the sequences of homologous genes and proteins can be identical, similar to varying degrees, or unrecognizably dissimilar because of extensive mutation.

Proteins and genes are either homologous or not, because they either did or did not descent from a common ancestor.

The only explanation other than divergence for similarities among sequences is **convergence**, in which two or more unrelated sequences have become similar under the pressure selection for similar functions. Convergence is a fairly common evolutionary phenomenon at the macroscopic level and is even encountered at the level of protein three-dimensional structure. There are, however, no instances in which nucleic acid or protein sequences have been shown conclusively to have become substantially similar by convergence. On the other hand, proving evolutionary convergence only from extant sequences is inherently difficult, so the possibility cannot be dismissed completely.

## Mutations

Mutations are both the raw material and the driving for of evolution, whereas **natural selection** modulates the rate of divergence.

A variety of mutations occur to DNA and RNA, by numerous and complex mechanisms. Simplest and most frequent is the replacement of one nucleotide by another; insertions and deletions of one or more nucleotides are also common. More complex rearrangements of DNA are less common but of greater consequence.

## Variation among species

The more closely related the organisms, the more similar the sequences of their genes and proteins are found to be. Closely related proteins generally differ only by replacement of one amino acid by another at a few positions in the polypeptide chain. Less frequent are differences in the total number of residues, which are due to the deletion or insertion of residues within or at either end of the polypeptide chain. In more distant relationships, the numbers and natures of sequence differences can increase greatly.

In general, the amino acid replacements that occur during protein divergence are non random, both in the extent to which various residues change and in the number of amino acids that replace each other. The most prevalent replacements occur between amino acids with similar side chains. This bias in amino acid replacements presumably reflects the role of selection; only those mutations that do not disrupt the function of the protein survive.

## Variation within species

DNA is known to be a very dynamic molecule that undergoes a wide variety of alterations and modifications, and **gene duplications** occur naturally and frequently. With two copies of a gene available in a genome, one copy could provide the necessary original function while the other accumulated mutations that altered its function. If this altered copy evolved eventually to serve a new function, it would tend to be retained in the genome and passed on to later generations.

Many genes and proteins of an organism are homologous and are obviously the products of gene duplication. The genes of such homologous proteins in a genome are said to comprise a gene family.

Genes that occupy the same gene locus in different species, and protein products of such genes, are said to be **orthologous**, whereas genes at different loci that are related by gene duplication are designated **paralogous** (see Figure 1). The phylogenies of species can be reconstructed only by comparing orthologous genes. The differences between paralogous genes or proteins from different species are not related to the time since divergence of the species, but to the time since gene duplication.

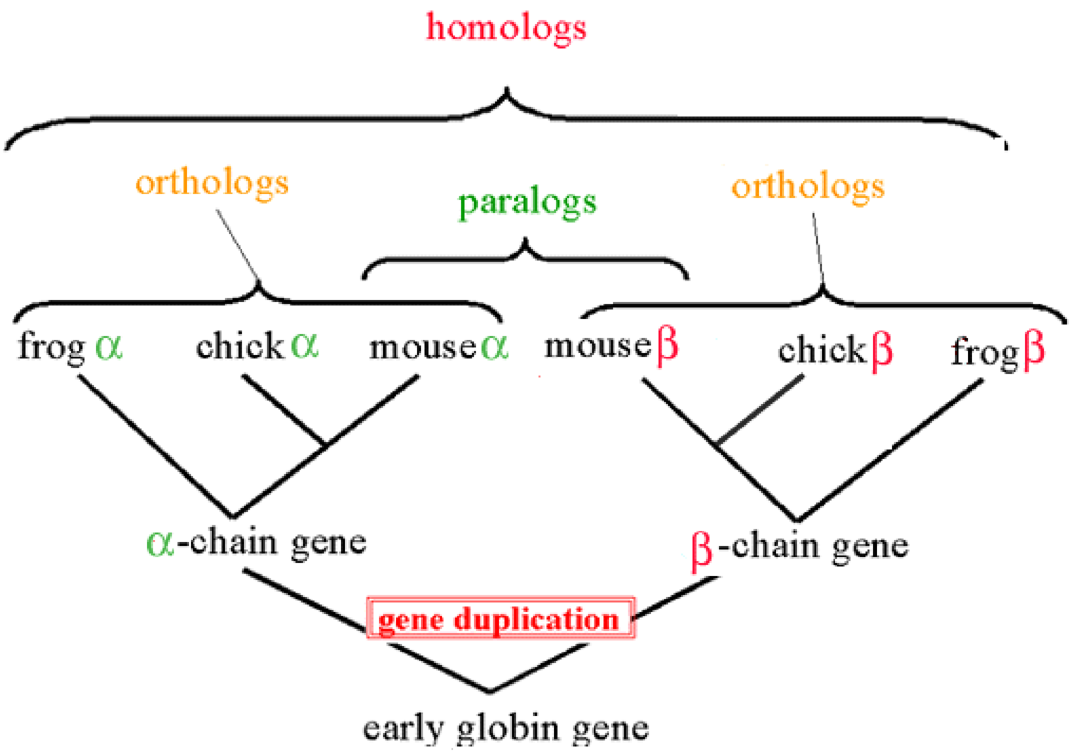


**Figure 1**: Illustration of homologous relationships: orthologs and paralogs.

## Domain shuffling

Many proteins, especially those unique to vertebrates, have mosaic structure in which various segments appear to have had different origins. Such a protein gives the impression of having been assembled by stringing modules together. Each module usually corresponds to an entire structural and functional domain of a protein.

Several different molecular mechanisms for domain shuffling have been proposed. Since the domains are often correlated with exon boundaries, exon shuffling is believed to be one of the major forces driving domain shuffling. Some other mechanisms might have been involved in domains shuffling, such as the simple fusion of genes and recruitment of mobile elements.

# Finding sequences to align

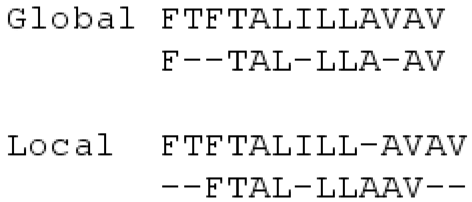
Very often the selection of sequences to align will be made using sequence similarity searches against a sequence database. The most commonly programs to perform these searches are the BLAST suite.

Because searches on a sequence database are perform using successive pair-wise alignments with the query sequence and each sequence in the database, it is convenient to revise some fundamentals concepts of pair-wise sequence alignments.

## Fundamentals: pair-wise sequence alignment

Let **a** = ($a_1$, . . . , $a_m$) and **b** = ($b_1$, . . . , $b_n$) be two sequences and a set of elementary operations including insertion, deletion, and substitution.

An alignment of **a** and **b** is a one-to-one correspondence such that each element of one sequence corresponds either to an element of the opposite sequence or to a null element indicating the presence of a gap.



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

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

There are two types of sequence alignment, global and local. In **global alignment**, an attempt is made to align the entire sequence, using as many residues, up to both ends of each sequence. Sequences that are quite similar and approximately the same length are suitable candidates for global alignment. In local alignment, stretches of sequence with the highest density of matches are aligned, thus generating one or more islands of matches or subalignments in the aligned sequences. **Local alignments** are more suitable for aligning sequences that are similar along some of their lengths but dissimilar in others, sequences that differ in length or sequences that share a conserved region or domain.

### Similarity scores

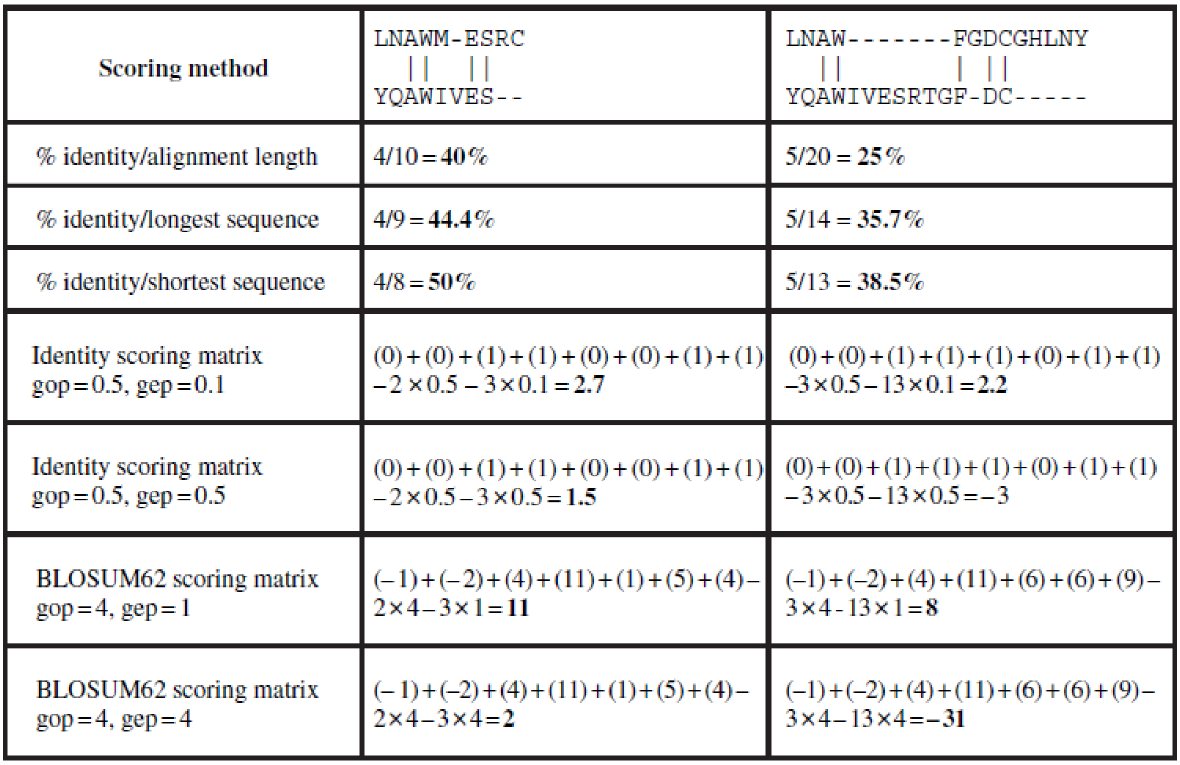
In contrast to homology, similarity is a quantitative measure and therefore we need to establish a numerical value from the sequence alignment. Many different methods have been proposed and used to address the question of an appropriate measure of similarity.

The simplest method involves counting the proportion of identical residues in aligned sequences relative to the alignment overall length, including gaps. This provides a percentage of identity that also takes into account the size of all gaps in the alignment.

A more complete method, like the ones used in most pair-wise alignments, computes a score for the sequence alignment by summing individual scores for stacked residues and subtracting a penalty for gaps:

- Individual scores for aligning residues are provided by **scoring matrices**, the simplest one being the identity matrix scoring 1 for identical residues and 0 otherwise. Many other matrices have also been designed to reflect amino acid properties. These replacement scores were either computed from physical and chemical properties or from observed frequencies of replacement of an amino acid by another in related proteins. Although real properties would seem to provide the most rational similarity scale, statistical scores actually reflect the effect of these properties on protein evolution and mutations allowed by natural selection. Statistical matrices eventually proved to be the most efficient ones and today, most similarity search programs use the statistical BLOSUM or PAM matrices built from reference alignments.
- The most widely used gap penalty is the so-called **affine gap penalty**. It is computed as a linear function of the number of gaps and their total length. Parameters provide control over the relative importance of number and length of gaps: a larger "gap opening penalty" will favor fewer but somewhat larger gaps, whereas a larger "gap extension penalty" would give preference to small gaps.

Some examples of different scoring methods are provided in figure 2 examples of alignment scores



| **Scoring method** | LNAWM-ESRC<br>\|\|   \|\|<br>YQAWIVES-- | LNAW-------FGDCGHLNY<br>\|\|       \|  \|\|<br>YQAWIVESRTGF-DC----- |
|---|---|---|
| % identity/alignment length | 4/10 = **40%** | 5/20 = **25%** |
| % identity/longest sequence | 4/9 = **44.4%** | 5/14 = **35.7%** |
| % identity/shortest sequence | 4/8 = **50%** | 5/13 = **38.5%** |
| Identity scoring matrix<br>gop = 0.5, gep = 0.1 | (0) + (0) + (1) + (1) + (0) + (0) + (1) + (1)<br>– 2 × 0.5 – 3 × 0.1 = **2.7** | (0) + (0) + (1) + (1) + (1) + (0) + (1) + (1)<br>–3 × 0.5 – 13 × 0.1 = **2.2** |
| Identity scoring matrix<br>gop = 0.5, gep = 0.5 | (0) + (0) + (1) + (1) + (0) + (0) + (1) + (1)<br>– 2 × 0.5 – 3 × 0.5 = **1.5** | (0) + (0) + (1) + (1) + (1) + (0) + (1) + (1)<br>– 3 × 0.5 – 13 × 0.5 = – 3 |
| BLOSUM62 scoring matrix<br>gop = 4, gep = 1 | (– 1) + (– 2) + (4) + (11) + (1) + (5) + (4) –<br>2 × 4 – 3 × 1 = **11** | (–1) + (–2) + (4) + (11) + (6) + (6) + (9) –<br>3 × 4 - 13 × 1 = **8** |
| BLOSUM62 scoring matrix<br>gop = 4, gep = 4 | (– 1) + (–2) + (4) + (11) + (1) + (5) + (4) –<br>2 × 4 – 3 × 4 = **2** | (–1) + (–2) + (4) + (11) + (6) + (6) + (9) –<br>3 × 4 – 13 × 4 = **–31** |

**Figure 2**: Examples of scores methods

### Scoring matrices

In the scoring matrices, also known as amino acid substitution matrices, amino acids are listed both across the top of a matrix and down the side, and each matrix position is filled with a score that reflects how often one amino acid would have been paired with the other in an alignment of related protein sequences.

All modern amino acid score matrices are estimated from frequencies observed in trusted alignment data, using some procedure to make a series of related matrices that are appropriate for different expected divergence.

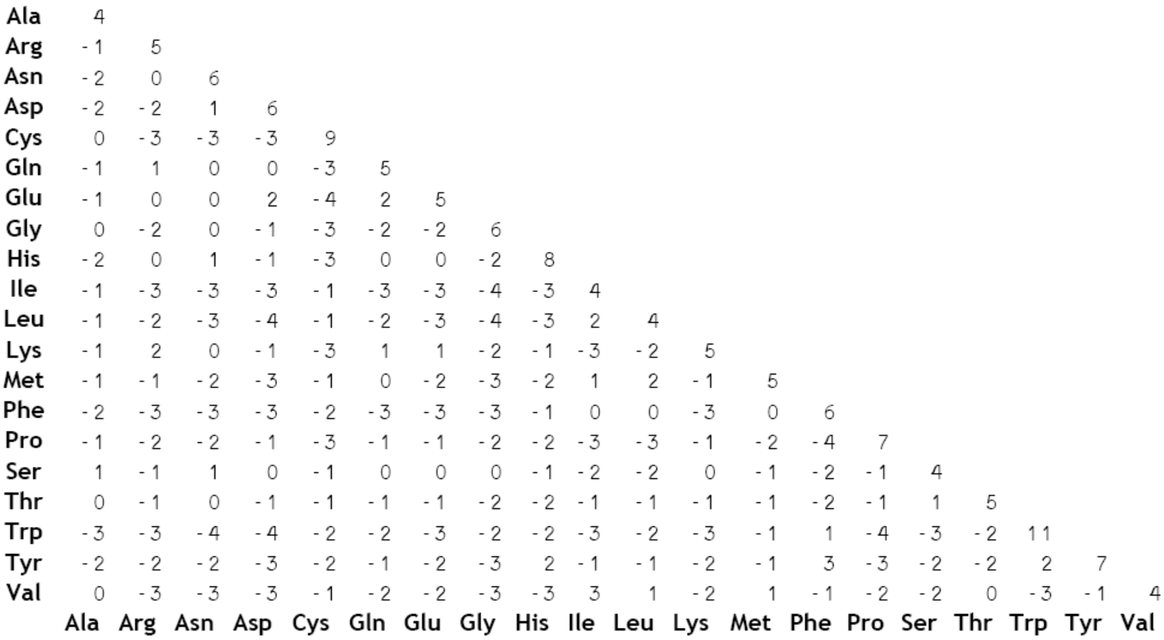
**PAM (Percent Accepted Mutation) matrices**

This family of matrices lists the likelihood of change from one amino acid to another in homologous protein sequences during evolution. Each matrix gives the changes expected for a given period of evolutionary time, evidenced by decreased sequence similarity as genes encoding the same protein diverge with increased evolutionary time.

The PAM matrices are normalized so that, for instance, the PAM1 matrix gives substitution probabilities for sequences that have experienced one point mutation for every hundred amino acids. The mutations may overlap so that the sequences reflected in the PAM250 matrix have experienced 250 mutation events for every 100 amino acids, yet only 80 out of every 100 amino acids have been affected.

This type of matrix is commonly known as a substitution matrix. Substitution matrices are used to derive scoring matrices used to assess the similarity of two aligned sequences. For example, an 18% probability of replacing arginine with lysine (in the substitution matrix) is turned into a score of 3 in the scoring matrix. The calculation uses the ratio of the probability value and the frequency of the original amino acid (arginine) in known sequences, known as the log-odds ratio.

**BLOSUM**



| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ala | 4 | | | | | | | | | | | | | | | | | | | |
| Arg | -1 | 5 | | | | | | | | | | | | | | | | | | |
| Asn | -2 | 0 | 6 | | | | | | | | | | | | | | | | | |
| Asp | -2 | -2 | 1 | 6 | | | | | | | | | | | | | | | | |
| Cys | 0 | -3 | -3 | -3 | 9 | | | | | | | | | | | | | | | |
| Gln | -1 | 1 | 0 | 0 | -3 | 5 | | | | | | | | | | | | | | |
| Glu | -1 | 0 | 0 | 2 | -4 | 2 | 5 | | | | | | | | | | | | | |
| Gly | 0 | -2 | 0 | -1 | -3 | -2 | -2 | 6 | | | | | | | | | | | | |
| His | -2 | 0 | 1 | -1 | -3 | 0 | 0 | -2 | 8 | | | | | | | | | | | |
| Ile | -1 | -3 | -3 | -3 | -1 | -3 | -3 | -4 | -3 | 4 | | | | | | | | | | |
| Leu | -1 | -2 | -3 | -4 | -1 | -2 | -3 | -4 | -3 | 2 | 4 | | | | | | | | | |
| Lys | -1 | 2 | 0 | -1 | -3 | 1 | 1 | -2 | -1 | -3 | -2 | 5 | | | | | | | | |
| Met | -1 | -1 | -2 | -3 | -1 | 0 | -2 | -3 | -2 | 1 | 2 | -1 | 5 | | | | | | | |
| Phe | -2 | -3 | -3 | -3 | -2 | -3 | -3 | -3 | -1 | 0 | 0 | -3 | 0 | 6 | | | | | | |
| Pro | -1 | -2 | -2 | -1 | -3 | -1 | -1 | -2 | -2 | -3 | -3 | -1 | -2 | -4 | 7 | | | | | |
| Ser | 1 | -1 | 1 | 0 | -1 | 0 | 0 | 0 | -1 | -2 | -2 | 0 | -1 | -2 | -1 | 4 | | | | |
| Thr | 0 | -1 | 0 | -1 | -1 | -1 | -1 | -2 | -2 | -1 | -1 | -1 | -1 | -2 | -1 | 1 | 5 | | | |
| Trp | -3 | -3 | -4 | -4 | -2 | -2 | -3 | -2 | -2 | -3 | -2 | -3 | -1 | 1 | -4 | -3 | -2 | 11 | | |
| Tyr | -2 | -2 | -2 | -3 | -2 | -1 | -2 | -3 | 2 | -1 | -1 | -2 | -1 | 3 | -3 | -2 | -2 | 2 | 7 | |
| Val | 0 | -3 | -3 | -3 | -1 | -2 | -2 | -3 | -3 | 3 | 1 | -2 | 1 | -1 | -2 | -2 | 0 | -3 | -1 | 4 |
| | Ala | Arg | Asn | Asp | Cys | Gln | Glu | Gly | His | Ile | Leu | Lys | Met | Phe | Pro | Ser | Thr | Trp | Tyr | Val |

**Figure 3**: BLOSUM62 scoring matrix

Henikoff & Henikoff took a big database of trusted alignments (the BLOCKS database), and counted pair-wise sequence alignments related by less than some threshold percentage identity. A threshold of 62% identity or less resulted in the target frequencies for the BLOSUM62 matrix. An 80% threshold gave the more highly conserved target frequencies of the BLOSUM80 matrix, and a 45% threshold gave the more divergent BLOSUM45 matrix. Empirically, the BLOSUM matrices have performed very well. BLOSUM62 has become a *de facto* standard for many protein alignment programs.

Generally speaking:

- The BLOSUM matrices are best for detecting local alignments
- BLOSUM62 is the best for detecting the majority of weak protein similarities (and the matrix used as default in most alignment programs)
- BLOSUM45 is the best for long and weak protein similarities.

**Table 1**: equivalences of PAM and BLOSUM matrices

| | |
|---|---|
| PAM100 | BLOSUM90 |
| PAM120 | BLOSUM80 |
| PAM160 | BLOSUM60 |
| PAM200 | BLOSUM52 |
| PAM250 | BLOSUM45 |

## Algorithms (Dynamic programming)

In mathematics and computer science, dynamic programming is a method for solving complex problems by breaking them down into simpler subproblems. The key idea is quite simple. In general, to solve a given problem, we need to solve different parts of the problem (subproblems), then combine the solutions of the subproblems to reach the overall solution. The dynamic programming approach seeks to solve each subproblem only once, thus reducing the number of computations.

Dynamic programming is a computational method that is used to align two protein or nucleic acid sequences. It provides the very best or **optimal alignment** between sequences. Both global and local types of alignments may be made by simple changes in the basic dynamic programming algorithm. A global alignment program is based on the Needleman-Wunsch algorithm, and a local alignment program on the Smith-Waterman algorithm.

Typically, the problem consists of transforming one sequence into another, using individual operations that either replace, insert, or remove an amino acid. Each operation has an associated score, as we have seen previously, and the goal is to find the sequence of individual operations with the highest total score.

The problem can be stated naturally as a recursion, a sequence **a** is optimally edited into a sequence **b** by either:

1. inserting the first character of **b**, and performing an optimal alignment of **a** and the tail of **b**
2. deleting the first character of **a**, and performing the optimal alignment of the tail of **a** and **b**
3. replacing the first character of **a** with the first character of **b**, and performing optimal alignments of the tails of **a** and **b**.

The partial alignments can be tabulated in a matrix, where cell (i,j) contains the cost of the optimal alignment of **a**[1..i] to **b**[1..j]. The score in cell (i,j) can be calculated by adding the score of the relevant operations to the score of its neighboring cells, and selecting the optimum.

## Basic Local Alignment Search Tool (BLAST)

Searching a sequence database for sequences that are similar to a query sequence is the most common type of database similarity search. The search provides a list of database sequences with which the query sequence can be aligned.

The dynamic programming (DP) method described above is guaranteed to find the highest-scoring alignment between two sequences. The time it takes to complete the alignment is proportional to the number of DP matrix elements to compute, which is the product of the sequence lengths. While this is very fast for comparing any two sequences of reasonable length, it is not practical for searching the current sequence databases, which contain many millions of sequences and many billions of residues. Therefore heuristic methods have been developed that can search entire databases much faster. While these methods do not guarantee finding the absolute best alignments, they have been finely optimized so that they have very high sensitivities and generally do obtain the optimal, or near-optimal, alignments. The most commonly used method is BLAST.

### The algorithm

1. The BLAST algorithm begins by fragmenting the sequence into 'words' (of 16-56 nucleotides, or 2-3 amino acids), and, from each word, creating a set of acceptable 'synonyms' that represent possible changes in sequence due to mutation.
2. Words and their synonyms are scored with respect to how well they match the query sequence, based on scoring matrices (e.g. BLOSUM). The words that match sufficiently well to have a score above a set threshold value are carried forward to compare to all the sequences in the database.
3. All the sequences in the database are then scanned for the presence of these words; sequences carrying two matches within a preset distance from each other are set aside until the entire database has been scanned. This "short list" of subject sequences is then carried forward by extending the alignment outward from the words to determine whether the match between the query and subject sequences extends beyond the local match between the subject sequence and the word. Initial "rough" alignments are extended without gaps to verify that the sequences match beyond the word hits. If the threshold score for the "ungapped" alignment is high enough that it suggests that the two sequences are indeed homologs, a second alignment is undertaken in which gaps are allowed to optimize the alignment. The sequences retrieved after these steps are referred to as the "subject" sequences.

It is important to note that alignment between the query and subject sequences does not have to cover the full length of the two sequences. Therefore, retrieved subject sequences commonly align with only a portion of the query sequence—it is this "local" rather than global quality that is more than nominally BLAST's strength.

Sometimes it is helpful to **mask** parts of the query sequence to prevent them from being aligned with subject sequences. Masking is helpful when the query sequence has low-complexity regions, such as stretches of small hydrophobic amino acids that are commonly present in transmembrane helices of integral membrane proteins. Because these features arose from convergent evolution, and their inclusion in BLAST searches could result in spurious hits, it is best to set the BLAST search parameters to eliminate these sorts of regions from word generation, as well as alignment scoring.

## Significance of an alignment score

The extend of the sequence similarity between the subject and query sequences is reported as a raw score, *S*

$$S = \left(\sum M_{ij}\right) - cO - dG$$

in which *M* is the score from a substitution matrix (e.g. BLOSUM62) for a particular pair of amino acids i and j, *c* is the number gaps, *O* is the penalty for the existence of a gap, *d* is the total length of the gaps, and *G* is the per-residue penalty for extending the gap.

Obtaining a score for an alignment does not, by itself, tell you whether it is significant. One needs to determine what is the probability of observing such a score by chance, given the scoring system used and the lengths of the sequences being compared.

**Bit scores:** Because one has the option of using different parameters (e.g. scoring matrices) in different BLAST searches, it is ideal to report results in such a way as to be able to compare alignments made with different scoring matrices or gap penalties. To do this, *S'* values (bit scores) are calculated.

$$S' = (\lambda S - \ln K) / \ln 2$$

in which λ and *K* depend on the matrices and penalties used.

**E-values**:

If analyses were to stop here, one would have a list of sequences sorted by bit scores that would reflect the degree of similarity to the query sequence. But how do we know if this similarity is significant, or is it only due to chance? (as it is the case of having a large database).

To address this issue E-values are calculated from bit scores as:

$$E = (n \times m) / \left(2^{S'}\right)$$

in which *n* is the total number of residues (amino or nucleic acids) in the database, and *m* is the length of the query sequence. E-values are the number of subject sequences that can be expected to be retrieved from the database that have a bit score equal to or greater than the one calculated from the alignment of the query and subject sequence, based on chance alone. E-values for subject sequences that are very similar to the query sequence will be quite small, and are widely used as a means to assess the confidence with which one should claim the subject sequence sequence(s) and the query sequence as homologs.

# Performing multiple sequence alignment

Most modern programs for constructing multiple sequence alignments (MSAs) consist of two components: an **objective function** for assessing the quality of a candidate alignment of a set of input sequences, and an **optimization procedure** for identifying the highest scoring alignment with respect to the chosen objective function.

In the case of multiple sequence alignment for N sequences, the multiple alignment score is usually defined to be the summed scores of all N(N – 1)/2 pair-wise projections of the original candidate MSA to each pair of input sequences. This is known as the standard sum-of-pairs (SP) scoring model. While other alternatives exist, such as consensus, entropy, or circular sum scoring, most alignment methods rely on the SP objective and its variants. Unlike the pair-wise case, direct dynamic programming methods cannot be applied to perform multiple sequence alignment under the SP scoring model, as these would require time and space exponential in N.

Therefore, most current techniques for SP-based multiple alignment work by either applying heuristics to solve the original NP-complete optimization problem approximately, or replacing the SP objective entirely with another objective whose optimization is tractable.

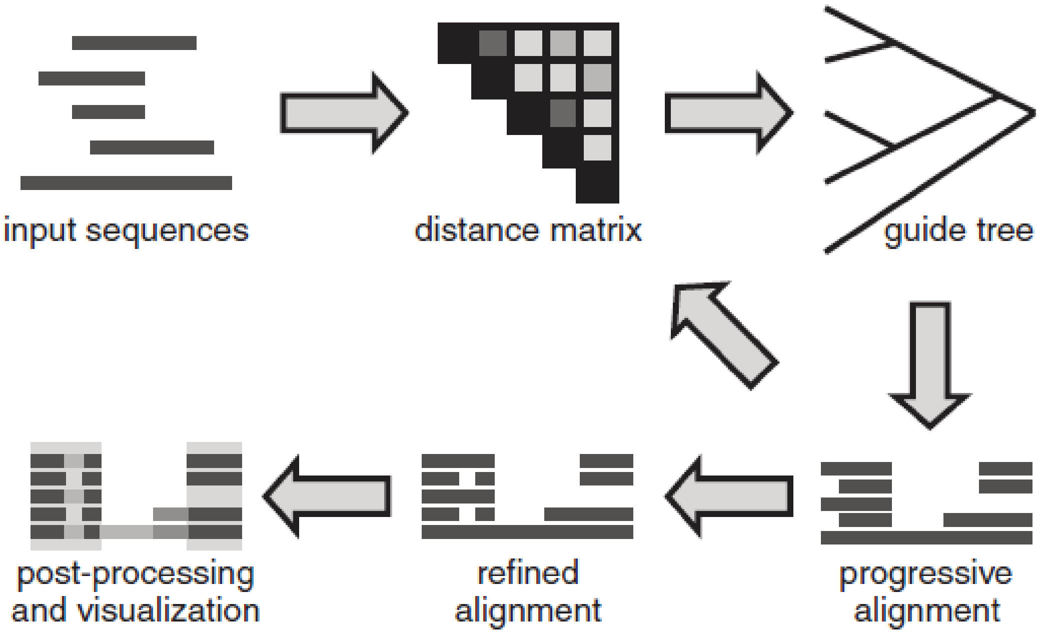


**Figure 4**: Diagram of the basic steps in a prototypical modern multiple sequence alignment program.

## Progressive methods

The most widely used approach to construct a multiple alignment is ‘progressive alignment' (see Figure XX), whereby a set of N proteins are aligned by performing N–1 pair-wise alignments of pairs of proteins or pairs of intermediate alignments, guided by a phylogenetic tree connecting the sequences. All progressive alignment methods require two stages: a first stage in which the relationships between the sequences are represented as a tree, called a *guide tree*, and a second step in which the MSA is built by adding the sequences sequentially to the growing MSA according to the guide tree. The initial guide tree is determined by an efficient clustering method.

### ClustalW

The most widely used programs for global multiple sequence alignment are from the Clustal series of programs.

ClustalW performs a global-multiple sequence alignment by the progressive method. The steps include:

a) Perform pair-wise alignment of all the sequences by dynamic programming

b) Use the alignment scores to produce a phylogenetic tree by neighbor-joining

c) Align the multiple sequences sequentially, guided by the phylogenetic tree

Thus, the most closely related sequences are aligned first, and then additional sequences and groups of sequences are added, guided by the initial alignments to produce a multiple sequence alignment showing in each column the sequence variations among the sequences.

Sequence contributions to the multiple sequence alignment are weighted according to their relationships on the predicted evolutionary tree. Weights are based on the distance of each sequence from the root. The alignment scores between two positions of the multiple sequence alignment are then calculated using the resulting weights as multiplication factors.

As more sequences are added to the profile, gaps accumulate and influence the alignment of further sequences. ClustalW calculates gaps in a novel way designed to place them between conserved domains. Gaps found in the initial alignments remain fixed. New gaps are then introduced into the multiple alignment when more sequences are added, but gaps can never be deleted, only added. ClustalW also implements methods, which try to compensate for the scoring matrix (e.g., PAM), expected number of gaps, and differences in sequence length.

### T-Coffee

Another common progressive alignment method called T-Coffee is slower than ClustalW and its derivatives but generally produces more accurate alignments for distantly related sequence sets. T-Coffee calculates pair-wise alignments by combining the direct alignment of the pair with indirect alignments that aligns each sequence of the pair to a third sequence. It uses the output from ClustalW as well as another local alignment program LALIGN, which finds multiple regions of local alignment between two sequences. The resulting alignment and phylogenetic tree are used as a guide to produce new and more accurate weighting factors.

### MUSCLE

MUSCLE (multiple sequence alignment by log-expectation) improves on progressive methods with a more accurate distance measure to assess the relatedness of two sequences. The distance measure is updated between iteration stages (although, in its original form, MUSCLE contained only 2-3 iterations depending on whether refinement was enabled).

## Iterative methods

A set of methods to produce MSAs while reducing the errors inherent in progressive methods are classified as "iterative" because they work similarly to progressive methods but repeatedly realign the initial sequences as well as adding new sequences to the growing MSA. One reason progressive methods are so strongly dependent on a high-quality initial alignment is the fact that these alignments are always incorporated into the final result - that is, once a sequence has been aligned into the MSA, its alignment is not considered further. This approximation improves efficiency at the cost of accuracy. By contrast, iterative methods can return to previously calculated pair-wise alignments or sub-MSAs incorporating subsets of the query sequence as a means of optimizing a general objective function such as finding a high-quality alignment score.

### MAFFT

MAFFT (multiple alignment using Fast Fourier Transform) is a multiple alignment program that offers a range of multiple alignment methods. Among them, two different heuristics are implemented: the progressive method (FFT-NS-2) and the iterative refinement method (FFT-NS-i).

**Table 2**: MSA software

| Tool | URL |
|---|---|
| CLUSTALW | www.clustal.org |
| DIALIGN | bibiserv.techfak.uni-bielefeld.de/dialign/ |
| MAFFT | mafft.cbrc.jp/alignment/software/ |
| MUSCLE | www.drive5.com/muscle/ |
| PRALINE | www.ibi.vu.nl/programs/pralinewww/ |
| PROBCONS | probcons.stanford.edu/ |
| ProDA | proda.stanford.edu/ |
| PROMALS | prodata.swmed.edu/promals/ |
| T-Coffee | www.tcoffee.org |

## Choosing the right MSA software

Given the multitude of choices, it can be difficult for a user of multiple alignment software to understand the situations in which a particular alignment tool is or is not appropriate. When aligning a small number (<20) of globally homologous sequences with high percent identity (>40%), most modern alignment programs will have no difficulty in returning a correct multiple sequence alignment, and no special consideration is needed. When all of these conditions do not hold, however, choosing the appropriate tools and configuration, while keeping in mind the tradeoff between accuracy and computational cost, can be difficult. You can find a list of currently popular alignment software (see Table 2) and advice on tool selection (see Fig. 5).

For more advanced discussion on software selection see (Do & Katoh, 2008).

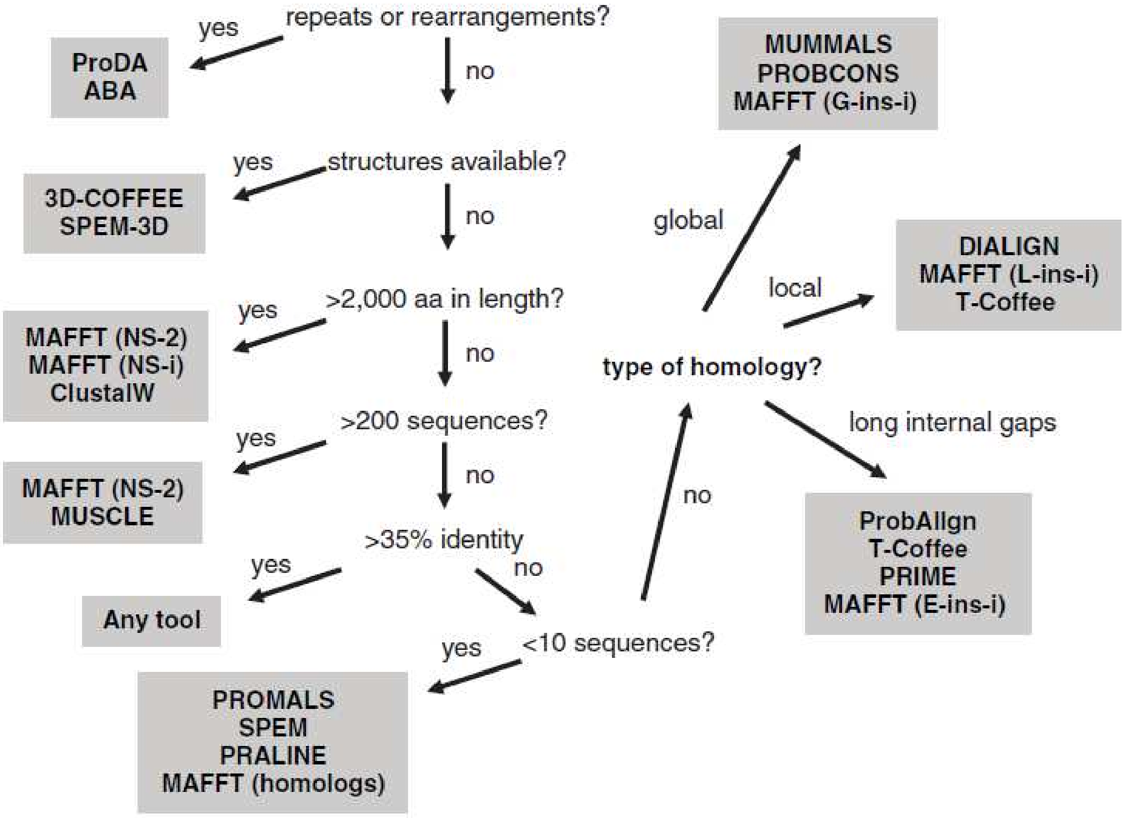


**Figure 5**: Decision tree for choosing the right MSA program.

## Alignment visualization and editing

Once an alignment has been generated, visualization tools allow manual identification of regions with reliably predicted homology; many of these tools also allow for interactive alignment editing.

For alignments of sequences with low similarity, post-processing is extremely important as most regions in a low-identity alignment will not be reliably alignable.

Visual inspection of an alignment and subsequent identification of potentially mis-aligned regions can be greatly helped using hints provided by software that highlight such regions. Typically, high confidence aligned regions can be identified by looking for groups of residues with strongly conserved physicochemical properties (e.g., hydropathy, polarity, and volume), using alternative alignment objective functions for identifying reliable columns, using posterior confidences generated by alignment programs such as PROBCONS, using the consensus of several alignment methods, or even better, cross-referencing aligned positions with amino acid residues in three-dimensional protein structures.

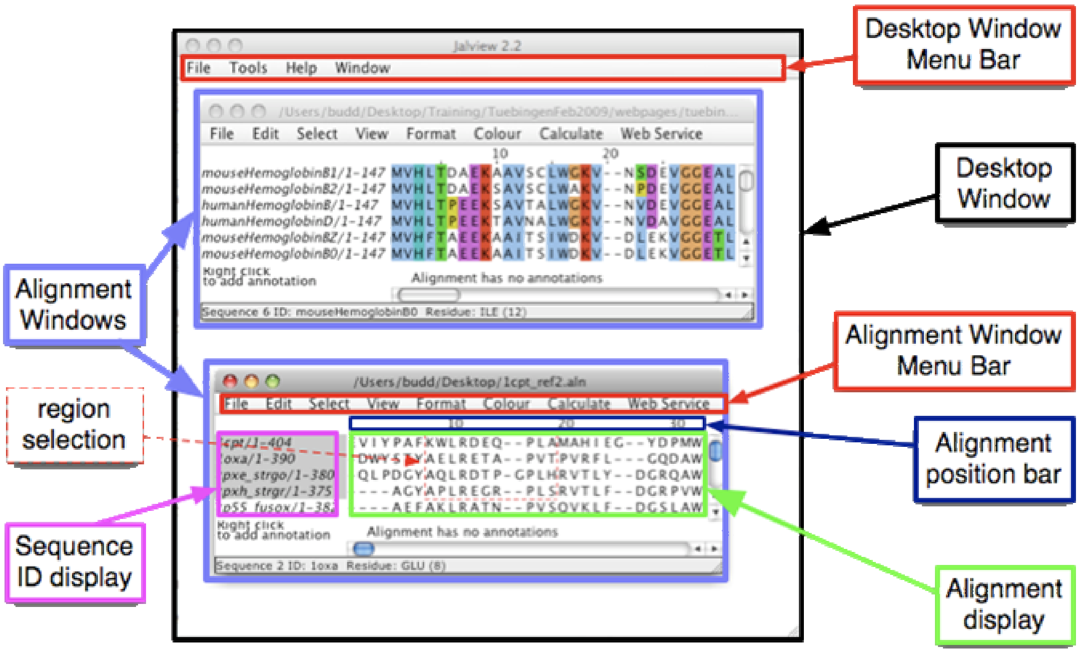
Once potential errors have been detected in alignment, they need to be corrected. While one can attempt to do this by simply editing the alignment file directly using a text editor, this is an error-prone approach instead several different pieces of software have been developed to carry out such operations interactively.

**Table 3**: Alignment visualization programs

| **Tool** | **URL** |
|---|---|
| Jalview | www.jalview.org |
| SeaView | pbil.univ-lyon1.fr/software/seaview.html |
| CINEMA | www.bioinf.manchester.ac.uk/dbbrowser/CINEMA2.1/ |
| STRAP | 3d-alignment.eu |
| ClustalX | www.clustal.org |
| ALTAVIST | bibiserv.techfak.uni-bielefeld.de/altavist/ |

## Jalview

Jalview is a multiple alignment editor written in Java. It is used widely in a variety of web pages (e.g. the EBI ClustalW server and the Pfam protein domain database) but is available as a general purpose alignment editor.


Desktop Window Menu Bar
Desktop Window
Alignment Windows
Alignment Window Menu Bar
region selection
Alignment position bar
Sequence ID display
Alignment display
Jalview 2.2
File Tools Help Window
/Users/budd/Desktop/Training/TuebingenFeb2009/webpages/tuebin...
File Edit Select View Format Colour Calculate Web Service
10
20
Right click to add annotation
Alignment has no annotations
Sequence 6 ID: mouseHemoglobinB0 Residue: ILE (12)
/Users/budd/Desktop/1cpt_ref2.aln
File Edit Select View Format Colour Calculate Web Service
10
20
30
Right click to add annotation
Alignment has no annotations
Sequence 2 ID: 1oxa Residue: GLU (8)

```
mouseHemoglobinB1/1-147 MVHLTDAEKAAVSCLWGKV--NSDEVGGEAL
mouseHemoglobinB2/1-147 MVHLTDAEKSAVSCLWAKV--NPDEVGGEAL
humanHemoglobinB/1-147  MVHLTPEEKSAVTALWGKV--NVDEVGGEAL
humanHemoglobinD/1-147  MVHLTPEEKTAVNALWGKV--NVDAVGGEAL
mouseHemoglobinBZ/1-147 MVHFTAEEKAAITSIWDKV--DLEKVGGETL
mouseHemoglobinB0/1-147 MVHFTAEEKAAITSIWDKV--DLEKVGGETL
```

```
1cpt/1-404     VIYPAFKWLRDEQ--PLAMAHIEG--YDPMW
1oxa/1-390     DWYSTYAELRETA--PVTPVRFL---GQDAW
pxe_strgo/1-380 QLPDGYAQLRDTP-GPLHRVTLY--DGRQAW
pxh_strgr/1-375 ---AGYAPLREGR--PLSRVTLF--DGRPVW
p55_fusox/1-382 ---AEFAKLRATN--PVSQVKLF--DGSLAW
```

**Figure 6**: Jalview interface

## Representing multiple sequence alignments

Although most of the time you will represent a multiple sequence alignment as a matrix of aligned residues, in some cases it is convenient to use alternative representations (as shortcuts or for generation of nice figures). Two of these alternative representations are the consensus sequences, and sequence logos. Alternative more complex representations will be presented in the next sections (Working with profiles).

### Consensus sequence

Consensus sequence refers to the most common nucleotide or amino acid at a particular position after multiple sequences are aligned. A consensus sequence is a way of representing the results of a multiple sequence alignment, where related sequences are compared to each other, and similar functional sequence motifs are found. The consensus sequence shows the residues that are most abundant in the alignment at each position.

Example: WW domain (from SMART database) represented as consensus sequences considering 80, 65 and 50%.

```
O54971/1-33      PLPPGWEKRTDSN-GRVYFVN---HNTRITQWEDPRS
CONSENSUS/80%    .h..sW..hhs.p.sh.aahs.....stpopWptPt.
CONSENSUS/65%    slsssWppthsss.GphYYhs...ppocpopWpcPp.
CONSENSUS/50%    sLPsGWccttsss.G+sYYaN...ppT+copW-cPss
```

The grouping of amino acids to classes and class abbreviation (the key) used within consensus sequences are shown below.

| Class | Key | Residues |
|---|---|---|
| alcohol | **o** | S,T |
| aliphatic | **l** | I,L,V |
| any | **.** | A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y |
| aromatic | **a** | F,H,W,Y |
| charged | **c** | D,E,H,K,R |
| hydrophobic | **h** | A,C,F,G,H,I,K,L,M,R,T,V,W,Y |
| negative | **-** | D,E |
| polar | **p** | C,D,E,H,K,N,Q,R,S,T |
| positive | **+** | H,K,R |
| small | **s** | A,C,D,G,N,P,S,T,V |
| tiny | **u** | A,G,S |
| turnlike | **t** | A,C,D,E,G,H,K,N,Q,R,S,T |

### Sequence logos

Sequence logos are a graphical way for presenting multiple alignments. The sequence logo shows how well residues are conserved at each position: the fewer the number of residues, the higher the letters will be, because the better the conservation is at that position. Different residues at the same position are scaled according to their frequency. The height of the entire stack of residues is the information measured in bits.

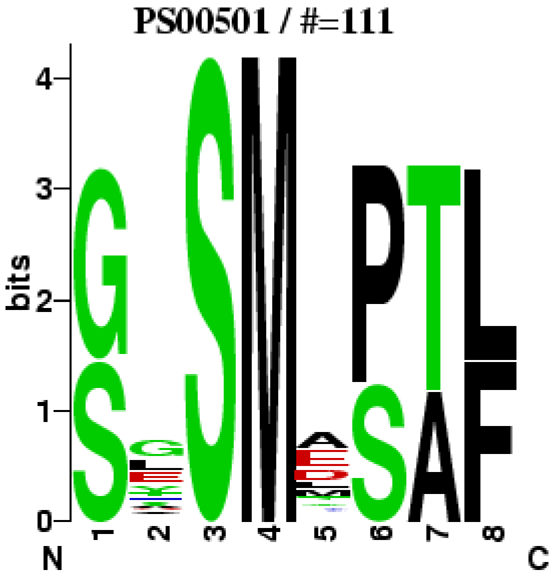


**Figure 7**: Sequence logo for the Signal peptidases I serine active site (PS00501) according to the PROSITE database.

# Working with profiles

When a number of member sequences of a protein family has been found, one can search for additional family members in at least two different ways. The first is to search with each known member against a sequence database. The second is to gather information from all known members, form a *model* for describing the properties of the members, and match this against a database of sequences. The latter has been shown to be superior in detecting weak relationships, i.e. remote family members. A number of different models has been applied. Each of these capture information about the family members and can be compared with sequences.
Most common models for the representation of domains and/or families are:

- **Sequence patterns** (motifs): these include consensus sequences and regular expressions.
- **Position-specific scoring matrices** (PSSM) or weight matrices. These are constructed from MSAs and represent the variation found in the alignment columns. It includes in addition to position-specific scores gap penalties to be used when comparing the profile to a sequence.
- **Hidden Markov Models** (HMM): these are conserved regions of multiple alignments represented as hidden markov models. The HMM is a statistical model that considers all possible combinations of matches, mismatches , and gaps to generate an alignment of a set of sequences.

Probabilistic models describing protein domains and families (e.g. PSSMs, HMMs) are globally known as **profiles** (in contrast to deterministic models such as sequence patterns).

## Sequence patterns

When a collection of diverse proteins shares a common function or structure, sometimes all that is conserved between them is a few common residues that are critical for their structure and function. If the proteins are enzymes, these residues are typically those that are involved in the chemical catalysis in the active site.

Sequence patterns are deterministic models: a sequence pattern is either matched or not matched by a sequence. The part of a sequence that actually matches a pattern is called an occurrence of the pattern. Patterns describing biologically meaningful similarities are called *motifs*.

These motifs, typically around 10 to 20 amino acids in length, arise because specific residues and regions thought or proved to be important to the biological function of a group of proteins are conserved in both structure and sequence during evolution. These biologically significant regions or residues are generally:

- Enzyme catalytic sites.

- Prostethic group attachment sites (heme, pyridoxal-phosphate, biotin, etc.).
- Amino acids involved in binding a metal ion.
- Cysteines involved in disulphide bonds.
- Regions involved in binding a molecule (ADP/ATP, GDP/GTP, calcium, DNA, etc.) or another protein

Different languages (or formalisms) for describing patterns exist, and for any language there is a mechanism for deciding whether or not a sequence matches a pattern. Here, we will focus on the PROSITE patterns.

### PROSITE patterns

PROSITE is a database of protein families and domains. The PROSITE language for sequential patterns is described by the following:

- the standard one-letter codes for amino acids are used
- the symbol 'x' is used for an arbitrary amino acid
- ambiguities are listed between square paranthenses '[ ]'. For example, [AGL] stands for Ala or Gly or Leu.
- Amino acids that are not accepted at a given position are listed between curly brackets'{ }'. For example, {CH} stands for any amino acid except Cys and His.
- '-' is used for separating the elements
- Repetition of an element is specified with a numerical value or a numerical range between parentheses, such that x(3) corresponds to x-x-x and x(1,3) corresponds to x or x-x or x-x-x.
- When a pattern is restricted to either the N- or C-terminal of a sequence, that pattern either starts with a '<' symbol or ends with a '>' symbol.
- A period ends the pattern

For example, the pattern corresponding to the Signal peptidases I serine active site (PS00501) is represented as

[GS]-{PR}-S-M-{RS}-[PS]-[AT]-[LF]

## Position-specific scoring matrices (PSSM)

The PSSM is constructed by a simple logarithmic transformation of a matrix giving the frequency of each amino acid in each position of a motif. Rows correspond to a position of the motif. The first 20 columns of each row specify the score for finding, at that position each of the 20 amino acid residues. An additional column contains a penalty for insertions or deletions at that position.

Two considerations arise in trying to tune the PSSM so that it adequately represents the training sequences. First, if the number of sequences with the found motif is large and reasonably diverse, the sequences represent a good statistical sampling of all sequences that are ever likely to be found with that same motif.

## Hidden Markov Models (HMM)

The HMM is a statistical model that considers all possible combinations of matches, mismatches, and gaps to generate an alignment of a set of sequences (see Figure 8).

The two HMM programs in common use are Sequence Alignment and Modeling Software System (SAM), and HMMER.

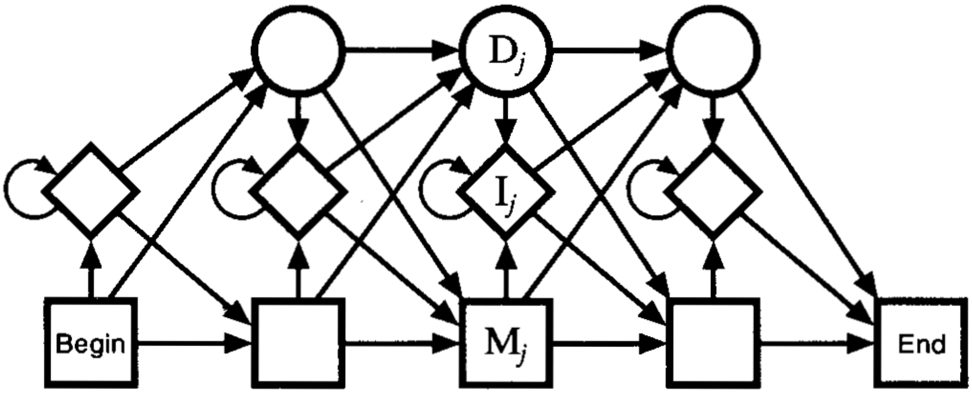


**Figure 8**: The structure of a Hidden Markov Model. Diamonds represent insert states and circles delete states.

## Domain/Family databases

There are a few databases available on the characterised protein families, domains and sites. These databases are constructed using different methodologies (e.g. sequence patterns, PSSMs, HMMs) and a varying degree of biological information.

These secondary protein sequence databases have become vital tools for identifying distant relationships in novel sequences and hence for inferring protein function. These databases have evolved by using signature-recognition methods to address different sequence analysis problems, resulting in rather different and independent databases (e.g. PROSITE, Pfam, PRINTS, etc).

While all of the resources share a common interest in protein sequence classification, some such as Pfam focus on divergent domains, some such as PROSITE focus on functional sites, and others such as PRINTS focus on families, specialising in hierarchical definitions from super-family down to sub-family levels in order to pin-point specific functions. A number of sequence cluster databases such as ProDom are also commonly used in sequence analysis, for example to facilitate domain identification.

Unfortunately, these secondary databases do not share the same formats and nomenclature as each other, which makes the use of all of them in an automated way difficult. In response to this the UniProtKB/Swiss-Prot group at the EBI has developed the “Integrated resource of Protein domains and functional sites” more commonly known as InterPro.

## Pfam

The Pfam database is a large collection of protein families, each represented by multiple sequence alignments and hidden Markov models.

The database comprises two main collections of information. Pfam-A comprises high-quality entries that have been curated manually. To extend the sequence coverage of Pfam, an additional area of the Pfam database, Pfam-B, contains automatically curated entries that are of a lower quality but add valuable coverage for regions not yet curated and stored in Pfam-A.

## SMART

SMART (a Simple Modular Architecture Research Tool) allows the identification and annotation of genetically mobile elements and the analysis of domain architectures. More than 500 domain families found in signalling, extracellular and chromatin-associated proteins are detectable. These domains are extensively annotated with respect to phyletic distributions, functional class, tertiary structures and functionally important residues. Each domain found in a non-redundant protein database as well as search parameters and taxonomic information are stored.

## InterPro

InterPro is an integrated documentation resource for protein families, domains, regions and sites. InterPro combines a number of databases (referred to as member databases) that use different methodologies and a varying degree of biological information on well-characterised proteins to derive protein signatures. By uniting the member databases, InterPro capitalises on their individual strengths, producing a powerful integrated database and diagnostic tool (InterProScan).

The member databases use a number of approaches:

1. ProDom: provider of sequence-clusters built from UniProtKB using PSI-BLAST.
2. PROSITE patterns: provider of simple regular expressions.
3. PROSITE and HAMAP profiles: provide sequence matrices.
4. PRINTS provider of fingerprints, which are groups of aligned, un-weighted Position Specific Sequence Matrices (PSSMs).
5. PANTHER, PIRSF, Pfam, SMART, TIGRFAMs, Gene3D and SUPERFAMILY: are providers of hidden Markov models (HMMs).

Diagnostically, these resources have different areas of optimum application owing to the different underlying analysis methods. In terms of family coverage, the protein signature databases are similar in size but differ in content. While all of the methods share a common interest in protein sequence classification, some focus on divergent domains (e.g., Pfam), some focus on functional sites (e.g., PROSITE), and others focus on families, specialising in hierarchical definitions from superfamily down to subfamily levels in order to pin-point specific functions (e.g., PRINTS). TIGRFAMs focus on building HMMs for functionally equivalent proteins and PIRSF always produce HMMs over the full length of a protein and have protein length restrictions to gather family members. HAMAP profiles are manually created by expert curators they identify proteins that are part of well-conserved bacterial, archaeal and plastid-encoded proteins families or subfamilies. PANTHER build HMMs based on the divergence of function within families. SUPERFAMILY and Gene3D are based on structure using the SCOP and CATH superfamilies, respectively, as a basis for building HMMs.

# Profile searches

Three types of searches can be performed with profiles:

1. Sequence-profile searches
2. Profile-sequence searches
3. Profile-profile searches

## Sequence-profile searches

Sequence-profile searches are typically performed when we use a query sequence to search a domain/family database (such as the databases included in the InterPro). Due to the different methodologies to build these databases, the underlying methods to perform the searches are also diverse.

The sequence-profile search in InterPro is performed by the InterProScan Sequence Search. InterProScan is a tool that combines the different protein signature recognition methods native to the InterPro member databases into one resource.

## Profile-sequence searches

These searches are performed when we use a query profile to search a sequence database. Typically, the profile will represent a domain or family, and we are interested in finding new sequences that contain the domain or are potential members of that family.

The profile is usually created from a multiple sequence alignment (e.g. HMMER) or automatically built by the search program (e.g. PSI-BLAST).

### HMMER

HMMER is used for searching sequence databases for homologs of protein sequences, and for making protein sequence alignments. It implements methods using Hidden Markov Models.

Compared to BLAST, FASTA, and other sequence alignment and database search tools based on older scoring methodology, HMMER aims to be significantly more accurate and more able to detect remote homologs because of the strength of its underlying mathematical models. In the past, this strength came at significant computational expense, but in the new HMMER3 project, HMMER is now essentially as fast as BLAST.

### PSI-BLAST

(Position-Specific Iterated BLAST) is a tool that produces a position-specific scoring matrix constructed from a multiple alignment of the top-scoring BLAST responses to a given query sequence. This scoring matrix produces a profile designed to identify the key positions of conserved amino acids within a motif. When a profile is used to search a database, it can often detect subtle relationships between proteins that are distant structural or functional homologues. These relationships are often not detected by a BLAST search with a sample sequence query.

In the first round, PSI-BLAST is just like a normal BLAST; it finds sequence homologues. In the second round or "iteration" of PSI-BLAST, it figures out which residues tend to be conserved by creating a custom profile for each position of the sequence from a multiple alignment. Then another BLAST is performed, using the profile to produce a position-specific scoring matrix based on which positions evolution has conserved vs. which positions evolution has allowed to vary. The sequences found after the first round are added to the profile, allowing PSI-BLAST to detect more distant homologues in each iteration.

## Profile-profile methods

Profile-profile searches are performed when we use a query profile to search against a profile database.

To improve the detection of related proteins, it is often useful to include evolutionary information for both the query and target proteins. One method to include this information is by the use of profile-profile alignments, where a profile from the query protein is compared with the profiles from the target proteins. Profile-profile alignments can be implemented in several fundamentally different ways.

# Content sources and bibliography

## Online resources

Course: “Making the most of your Multiple Sequence Alignments (MSAs)” Aidan Budd and David Judge. March 2007. Organised by Depatment of Genetics, Cambridge University and EMBL (European Molecular Biology Laboratory)
http://www.embl.de/~seqanal/MSAcambridgeGenetics2007/

European Bioinformatics Institute (EBI):
http://www.ebi.ac.uk/help/matrix.html
http://www.ebi.ac.uk/interpro/user_manual.html

Folding@Home
http://www.stanford.edu/group/pandegroup/folding/education/h.html

HMMER http://hmmer.janelia.org

JalView http://www.jalview.org

Pfam database:
http://pfam.sanger.ac.uk/

PSSM viewer (NCBI):
http://www.ncbi.nlm.nih.gov/Class/Structure/pssm/pssm_viewer.cgi

SMART database:
http://smart.embl-heidelberg.de

Using ClustalX for multiple sequence alignment. Jarno Tuimala

Weblogos
http://weblogo.berkeley.edu/examples.html

Wikipedia:
http://en.wikipedia.org

## Books and articles

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. *J Mol Biol.* 1990 215(3):403-10.

Claverie JM & Notredame C. Bioinformatics for dummies. Wiley Publishing Inc. 2007

Do CB, Katoh K. Protein multiple sequence alignment. *Methods Mol Biol.* 2008 484:379-513.

Durbin R, Eddy SR, Krogh A, Mitchison G. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press. 1998

Henikoff S & Henikoff JG. Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A*. 1992 89(22):10915-9.

Kawashima T et al. Domain shuffling and the evolution of vertebrates. *Genome Res*. 2009 19:1393-1403.

Kerfeld CA & Scott KM. Using BLAST to teach “E-value-tionary” concepts. *PLoS Biol*. 2011 9(2):e1001014.

Mount DW. Bionformatics – Sequence and genome analysis. Cold Spring Harbor Laboratory Press. 2004

Needleman SB & Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*. 1970 48(3):443-53.

Ohlson T, Wallner B, Elofsson A. Profile-profile methods provide improved fold-recognition: a study of different profile-profile alignment methods. *Proteins*. 2004 57(1):188-97.

Pál C, Papp B, Lercher MJ. An integrated view of protein evolution. *Nat Rev Genet*. 2006 7(5):337-48.

Plewniak F. Database similarity searches. *Methods Mol Biol*. 2008 484:361-78.

Smith TF & Waterman MS. Identification of common molecular subsequences. *J Mol Biol*. 1981 147(1):195-7.