



Bioinformatics explained: Multiple Sequence Alignments

Multiple alignments are at the core of bioinformatical analysis. Often the first step in a chain of bioinformatical analyses is to construct a multiple alignment of a number of homologous nucleotide or protein sequences. However, despite their frequent use, the development of multiple alignment algorithms remains one of the algorithmically most challenging areas in bioinformatical research.

Constructing a multiple alignment corresponds to developing a hypothesis of how a number of sequences have evolved through the processes of character substitution, insertion and deletion. The input to multiple alignment algorithms is a number of homologous sequences, i.e. sequences that share a common ancestor and most often also share common molecular functions. The generated alignment is a table (see figure 1) where each row corresponds to an input sequence, and each column corresponds to a position in the alignment. An individual column in this table represents residues that have all diverged from a common ancestral residue. Gaps in the table (commonly represented by a '-') represent positions where residues have been inserted or deleted and thus do not have ancestral counterparts in all sequences.

Use of multiple alignments

Once a multiple alignment is constructed it can form the basis for a number of analyses:

- The phylogenetic relationship of the sequences can be investigated by tree-building methods based on the alignment.
- Annotation of functional domains, which may only be known for a subset of the sequences, can be transferred to aligned positions in other un-annotated sequences.
- Conserved regions in the alignment can be found which are prime candidates for holding functionally important sites.



Figure 1: The tabular format of a multiple alignment of 24 Hemoglobin protein sequences. Sequence names appear at the beginning of each row and the residue position is indicated by the numbers at the top of the alignment columns. The level of sequence conservation is shown on a color scale with blue residues being the least conserved and red residues being the most conserved.

The interpretation of a multiple sequence alignment can, as mentioned above, act as the basis of analysis of evolutionary relationship. The alignment can be used to search for homology between

sequences or groups of sequences, and mutations (insertions, deletions or point mutation) can be detected. Furthermore, a sequence alignment can indicate structural and/or functional characteristics of sequences and in comparison with well-described sequences new information may be gained about so far unknown sequence data. Conserved domains, which may indicate functionally important sites as e.g. binding sites, active sites or sites related to other key functions, can be identified by conserved regions in the sequence alignment.

There are different ways of visual inspection of an alignment. As shown and described with figure 1 the level of sequence conservation can typically be shown on a color scale, and depending on the interface used for generating the alignment, a number of other possibilities may be available. Conservation may be shown not only as a color scale on the alignment residues but also as a graph below the alignment to easily detect either inconsistencies or completely conserved regions. It is also common practice to represent a consensus sequence displaying the most frequent residues at each given position in the alignment. The consensus sequence may be determined by reflecting majority among the sequences, by a 100% match or by any other option available from the alignment program.

It may also be relevant to consider different aspects of the sequence comparison as for instance if the sequences should be compared according to identity, similarity or homology. Identity relates to sequences containing identical residues at correlated positions, whereas similarity relates to a quantitative statement of residues *similar* to each other. For nucleotide sequences, pyrimidines are considered to be similar as well as purines are considered similar. For protein sequences, similarity is understood as the means of similar physiochemical characteristic of the proteins. Homology in sequence alignments depends on similarity - the more similar the sequences are, the closer they are to being homologous. The homology of sequences may indicate an evolutionary origin and relates thus to a qualitative statement.

Constructing multiple sequence alignments

Whereas the optimal solution to the pairwise alignment problem can be found within reasonable time, the problem of constructing a multiple alignments is much harder and much more complex. To minimize difficulties as well as time, heuristic algorithms are used for construction and analysis of multiple sequence alignments.

The first major challenge in the multiple alignment procedure is how to rank different alignments, i.e. which *scoring function* to use. Since the sequences have a shared history they are correlated through their *phylogeny*, and the scoring function should ideally take this into account. Doing so is, however, not straightforward as it increases the number of model parameters considerably. It is therefore commonplace to either ignore this complication and assume sequences to be unrelated, or to use heuristic corrections for shared ancestry.

The second challenge is to find the optimal alignment given a scoring function. For pairs of sequences this can be done by *dynamic programming* algorithms, identifying global optimal alignments. This method is, however, only used for a small number of extremely homologous sequences being aligned as the approach otherwise demands too much computer time and memory to be feasible. Dynamic programming handling n sequences requires the construction of the n -dimensional equivalent of the matrix formed in standard pairwise dynamic programming, and the need for search space will thus exponentially increase with increasing n .

Heuristic methods start determining pairwise alignments of the most closely related sequences,

progressively adding less related sequences to the initial alignment. Because of the relatively limited requirements for time and power compared to global optimization methods, heuristic methods are widely used in multiple sequence alignment programs. A commonly used approach is therefore to do *progressive alignment* [Feng and Doolittle, 1987], where multiple alignments are built through the successive construction of pairwise alignments, thus being useful for a higher number of distantly related sequences. Progressive methods are very dependent on the initial alignment and thus more likely to perform well for closely related sequences, which may be seen as a limitation of the method. On the other hand, progressive methods provide a good compromise between time spent and quality of the resulting alignment.

The most widely used alignment algorithm for multiple sequence alignments, ClustalW [Thompson et al., 1994], is built on progressive alignment. The progressive technique is also used to optimize e.g. the T-Coffee alignment method [Notredame et al., 2000].

Iterative methods work similarly to progressive methods but improve the progressive alignment by continuously considering the existing alignment when adding new sequences. Iterative methods thus repeatedly realign the initial pairwise alignment, in the progressive alignment method, to new sequences being added. This improves the accuracy of the popular progressive methods. The MUSCLE alignment algorithm is based on iterative methods [Edgar, 2004].

Presently, the most exciting development in multiple alignment methodology is the construction of *statistical alignment* algorithms [Hein, 2001], [Hein et al., 2000]. These algorithms employ a scoring function which incorporates the underlying phylogeny and use an explicit stochastic model of molecular evolution, which makes it possible to compare different solutions in a statistically rigorous way. The optimization step, however, still relies on dynamic programming, and practical use of these algorithms thus awaits further developments.

Multiple sequence alignment methods

A number of different algorithms are available for construction of multiple sequence alignments. This section shortly describes three of the most commonly used alignment algorithms based on some of the above mentioned considerations.

ClustalW

Algorithms within the Clustal family are built on the progressive alignment method. ClustalW is a weighted variant introduced in 1994 improving the progressive approach for aligning distantly related protein sequences [Edgar and Batzoglou, 2006], [Thompson et al., 1994]. Improvements relate to the introduction of weighted gap penalties, use of different weight matrices in different stages of the alignment process, and weighting of sequences according to divergency.

ClustalW is one of the most popular alignment methods and the method is accessible from several web services, e.g. GenomeNet (<http://align.genome.jp/>) and the European Bioinformatics Institute, EBI (<http://www.ebi.ac.uk/clustalw/>).

T-Coffee

The alignment method T-Coffee is also based on the progressive method. The T-Coffee algorithm uses Clustal and combines direct and indirect alignments to find more accurate alignments when handling distantly related sequences. Data sets of all possible pairwise alignments between the sequences of interest are prepared to direct the progressive alignment process. Progressing the alignment indirectly aligns each of the sequences in the direct alignment to any third sequence



considered in the calculation. With T-Coffee the multiple sequence alignment of divergent sequences is greatly improved according to accuracy [Notredame et al., 2000]. Due to the improvements on accuracy, T-Coffee may not compare to ClustalW on speed.

MUSCLE

MUSCLE (multiple sequence alignment by log-expectation) is a popular iteration-based alignment method using *k*mer counting for estimation of distance and a log-expectation score as profile function. MUSCLE improves both accuracy and speed compared to other well-known and accepted methods for multiple sequence alignments (e.g. T-Coffee and ClustalW, respectively) [Edgar, 2004].

Table 1
Summary of selected multiple sequence alignment programs

Program	Advantages	Cautions
ClustalW	Uses less memory than other programs	Less accurate or scalable than modern programs
MUSCLE	Faster and more accurate than ClustalW; good trade-off of accuracy and computational cost. Options to run even faster, with lower average accuracy, for high-throughput applications	For very large data sets (say, more than 1000 sequences) select time- and memory-saving options
T-Coffee	High accuracy and the ability to incorporate heterogeneous types of information	Computation time and memory usage is a limiting factor for large alignment problems (>100 sequences)

Adapted from Edgar and Batzoglou, 2006

Biological accuracy, execution time and memory usage are three main considerations when selecting a multiple sequence alignment algorithm. Accuracy should get the highest priority considering scientific interests, and ClustalW is probably the most popular tool for creating multiple sequence alignments due to the reasonable balance between accuracy and computational costs required. A short summary of advantages and cautions of the three alignment methods described above is shown in table 1.

As multiple sequence alignments are of great importance in bioinformatics and as the amount of sequence data to be analyzed is highly increasing, methods are still being improved and developed with the purpose of reducing memory usage and increasing speed as well as improving the quality of the alignments [Edgar and Batzoglou, 2006].

Scoring matrices and gap penalties

The alignment methods described above are built on algorithms considering different parameters to align multiple sequences and determine their similarity.

Each column in the alignment can be assigned a 'cost' in order to calculate the cheapest alignment and thus the optimal alignment. The sum of pairs method calculates sum of pairs for each column in the alignment and sum all sum of pairs scores to get the total score of the alignment. To use this method all possible alignments has to be taken into account. This is time consuming and complex and may require a lot of computational power as well as time. This is dealt with by the progressive alignment methods reducing the complexity of multiple sequence alignments to a series of pairwise alignments [Edgar and Batzoglou, 2006].

Nucleotide alignments have four standard characters available for each position in the alignment, and the parameters for DNA and RNA sequence alignments would typically be gap penalties and

The PAM matrix, developed as one of the first substitution matrices back in the 1970s is derived from estimation of mutation rates in global alignments of closely related proteins. PAM1 matrix is calculated from protein alignments of sequences only differing by 1%, and PAM1 is used as the basis for other matrices within the PAM family for use with more divergent sequences [Dayhoff et al., 1978].

BLOSUM matrices have the probability scores computed from protein blocks of related sequences i.e. from local alignments. From more than 2000 blocks, each representing a conserved region within a protein family, around 500 groups of related proteins have been characterized and a number of blocks substitution matrices for use according to sequence similarity have been constructed from this. High number BLOSUM matrices is best suited for alignment of closely related sequences while low number BLOSUM matrices should be the choice for comparing distantly related sequences [Henikoff and Henikoff, 1992].

Calculation of alignment costs considering gap penalties and scoring matrices

To identify the optimal sequence similarity, the cheapest alignment of sequences has to be found, and the cost of any possible alignment within the sequence comparison has to be calculated.

For two sequences, Sequence $A = (1 \dots i)$ and Sequence $B = (1 \dots j)$ the cost(i,j) of alignment is calculated by a score for residue substitution and a gap penalty.

The comparison of the two sequences $A = (1 \dots i)$ and $B = (1 \dots j)$ can be described as

$$S_{ij} = \min \begin{cases} S_{i-1,j} + g \\ S_{i,j-1} + g \\ S_{i-1,j-1} + W(A[i], B[j]) \end{cases}$$

where g is the gap cost and $w(A[i], B[j])$ is the cost of substituting residue $A[i]$ with $B[j]$.

First and second term consider the alignment by inserting a gap into sequence A or sequence B , respectively, extending the two sequences compared by one residue each.

Third term considers an extension of the alignment by extending the two sequence compared by one residue each.

For nucleotide sequence alignments the substitution cost will be assigned a value for match/mismatch whereas for protein sequence alignments, the higher complexity of substitutions requires a more advanced scoring matrix, taking into account the different substitution possibilities between protein sequences. Similarity of residues A_i and B_j is then given by a weight matrix considering matches, substitutions or insertions/deletions.

Other useful resources

Wikipedia on Multiple Sequence Alignments

http://en.wikipedia.org/wiki/Multiple_sequence_alignment

ExpASy sequence alignment tools

<http://www.expasy.org/tools/#align>

ClustalW

<http://www.ebi.ac.uk/clustalw/>

Muscle

<http://www.drive5.com/muscle/>

T-Coffee

<http://www.tcoffee.org/>

Creative Commons License

All CLC bio's scientific articles are licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 2.5 License. You are free to copy, distribute, display, and use the work for educational purposes, under the following conditions: You must attribute the work in its original form and "CLC bio" has to be clearly labeled as author and provider of the work. You may not use this work for commercial purposes. You may not alter, transform, nor build upon this work.



See <http://creativecommons.org/licenses/by-nc-nd/2.5/> for more information on how to use the contents.

References

- [Dayhoff et al., 1978] Dayhoff, M. O., Schwartz, R. M., and Orcutt, B. C. (1978). A model of evolutionary change in protein. *Atlas of Protein Sequence and Structure*, 5(3):345–352.
- [Edgar, 2004] Edgar, R. C. (2004). Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*, 32(5):1792–1797.
- [Edgar and Batzoglou, 2006] Edgar, R. C. and Batzoglou, S. (2006). Multiple sequence alignment. *Curr Opin Struct Biol*, 16(3):368–373.
- [Feng and Doolittle, 1987] Feng, D. F. and Doolittle, R. F. (1987). Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J Mol Evol*, 25(4):351–360.
- [Hein, 2001] Hein, J. (2001). An algorithm for statistical alignment of sequences related by a binary tree. In *Pacific Symposium on Biocomputing*, page 179.
- [Hein et al., 2000] Hein, J., Wiuf, C., Knudsen, B., Møller, M. B., and Wibling, G. (2000). Statistical alignment: computational properties, homology testing and goodness-of-fit. *J Mol Biol*, 302(1):265–279.
- [Henikoff and Henikoff, 1992] Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A*, 89(22):10915–10919.
- [Notredame et al., 2000] Notredame, C., Higgins, D. G., and Heringa, J. (2000). T-coffee: A novel method for fast and accurate multiple sequence alignment. *J Mol Biol*, 302(1):205–217.
- [Thompson et al., 1994] Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994). Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, 22(22):4673–4680.