

4 Multiple Sequence Alignment

Sources for this lecture:

- R. Durbin, S. Eddy, A. Krogh und G. Mitchison, Biological sequence analysis, Cambridge, 1998
- D. Gusfield, Algorithms on string, trees and sequences, 1997.
- D.W. Mount. Bioinformatics: Sequences and Genome analysis, 2001.
- J. Setubal & J. Meidanis, Introduction to computational molecular biology, 1997.
- M. Waterman. Introduction to computational biology, 1995.

4.1 Multiple sequence alignment (MSA)

A multiple sequence alignment (MSA) is simply an alignment of more than two sequences, like this:

```
MRP2_HUMAN      TSNRWLAIRLELVGNLTVFFSALMMVIY--RDTLSGDTVGFVLSNALNITQTLNWLVRMT
Q9UQ99_HUMAN    VANRWLAVRLECVGNCIVLFAALFAVIS--RHSLSAGLVGLSVSYSLQVTTYLNWLVRMS
ABCC8_HUMAN     AANRWLEVRMEYIGACVVLIAAVTSISNSLHRELSAGLVGLGLTYALMVSNYLNMVMVRNL
Q96J65_HUMAN    CALRWFALRMDVLMNLTFTVALLVTLS--FSSISTSSKGLSLSYIIQLSGLLQVCVRTG
Q96JA6_HUMAN    SSTRWMALRLEIMTNLVTALAVALFVAFG--ISSTPYSPFKVMVNIQLASSFQATARIG
MRP5_HUMAN      CAMRWLAVRDLISIALITTTGLMIVLM--HGQIPPAYAGLAISYAVQLTGLFQFTVRLA
MRP4_HUMAN      TTSRWFAVRLDAICAMFVIIVAFGSLIL--AKTLDAGQVGLALSALYALTMGMFQWCVRQS
O75555_HUMAN    TTSRWFAVRLDAICAMFVIIVAFGSLIL--AKTLDAGQVGLALSALYALTMGMFQWCVRQS
CFTR_HUMAN      STLRFQFMRIEMIFVIFFIIVTFISILT---TGEGERGVGIILTAMNIMSTLQWAVNSS
```

(A small section of a multiple alignment of the human CFTR protein and eight homologous proteins.)

Multiple sequence alignment is applied to a set of sequences that are assumed to be related and the goal is to detect homologous residues and to place them in the same column of the multiple alignment. Multiple alignments are more suitable than pairwise alignments to address evolutionary questions, as the chance of random similarities occurring decreases, as the number of aligned sequences grows.

Quote (Arthur Lesk): One or two homologous sequences whisper ... a full multiple sequence alignment shouts out loud...

4.1.1 Characterization of protein families

Typical question: Suppose that $F = \{A_1, A_2, \dots, A_r\}$ is an established family of homologous protein sequences. Does some new sequence A_0 belong to the family?

One way to address this question would be to align A_0 to each of A_1, \dots, A_r in turn. If any one of these alignments produces a high score, then we may decide that A_0 belongs to the family F .

However, perhaps A_0 does not align particularly well to any *one* specific family member, but scores well in a *multiple* alignment, perhaps due to a common motif or conserved feature.

4.1.2 Conservation of structural elements

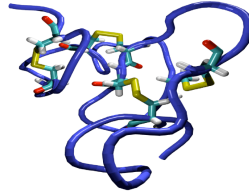
For example, here is an alignment of N-acetylglucosamine-binding proteins:

```

AATAHAQRCG EQGSNMECPN NLCCSQYGYC GMGGDYCGKG ..CQNGACYT
VAATNAQTCG KQNDGMIAPH NLCCSQFGYC GLGRDYCGTG ..CQSGACCS
VGLVSAQRCG SQGGGGTTPA LWCCSIWGWG GDSEPYCGRT ..CENK.CWS
AATAHAQRCG EQGSNMECPN NLCCSQYGYC GMGGDYCGKG ..CQNGACWT
AATAHAQRCG EQGSNMECPN NLCCSQYGYC GMGGDYCGKG ..CQNGACWT
.....QRCG EQGSNMECPN NLCCSQYGYC GMGGDYCGKG ..CQNGACWT
SETVKSQNCG .....CAP NLCCSQFGYC GSTDAYCGTG ..CRSGPERS
RGSAAE..QCG RQAGDALCPG GLCCSFYGWG GTTVDYCGDG ..CQSQ.CDG
AGPAAQNCG .....CQP NFCCSKFGYC GTTDAYCGDG ..CQSGPERS
AGPAAQNCG .....CQP NVCCSKFGYC GTTDEYCGDG ..CQSGPERS
RGSAAE..QCG RQAGDALCPG GLCCSFYGWG GTTADYCGDG ..CQSQ.CDG
RGSAAE..QCG RQAGDALCPG GLCCSFYGWG GTTVDYCGDG ..CQSQ.CDG
TGVAIAEQCG RQAGGKLC PN NLCCSQWGWG GSTDEYCPD HNCQSN.CK.
.....EQCG RQAGGKLC PN NLCCSQWGWG GSDDYCSPS KNCQSN.CK.

```

The alignment shows 8 conserved cysteins. These form 4 disulfide bridges, which are essential to stabilize the protein:



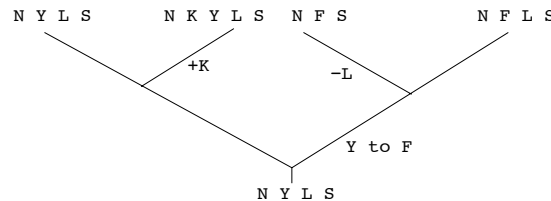
A new sequence should only be added to the family if it has this pattern of conserved cysteins, which can be seen using a multiple sequence alignment.

4.1.3 MSA and evolutionary trees

Another main application of multiple sequence alignments is in phylogenetic analysis. Suppose we are given an MSA, e.g.:

$$\begin{aligned}
 A_1^* &= N - F L S \\
 A_2^* &= N - F - S \\
 A_3^* &= N K Y L S \\
 A_4^* &= N - Y L S
 \end{aligned}$$

We would like to reconstruct the evolutionary tree that gave rise to these sequences, e.g.:



The computation of phylogenetic trees will be discussed in a later chapter.

4.2 Definition of an MSA

Suppose we are given r sequences $A_i, i = 1, \dots, r$:

$$\mathbf{A} = \begin{cases} A_1 = a_{11} & a_{12} & \dots & a_{1n_1} \\ A_2 = a_{21} & a_{22} & \dots & a_{2n_2} \\ & \vdots & & \\ A_r = a_{r1} & a_{r2} & \dots & a_{rn_r} \end{cases}$$

Definition 4.2.1 (Multiple sequence alignment (MSA)) A multiple sequence alignment \mathbf{A}^* of \mathbf{A} is obtained by inserting gaps ('-') into the original sequences such that:

1. All resulting sequences A_i^* have equal length $L \geq \max\{n_i \mid i = 1, \dots, r\}$,
2. Removal of all gaps in A_i^* produces A_i , and

3. No column consists only of gaps.

$$\mathbf{A}^* = \begin{cases} A_1^* = a_{11}^* & a_{12}^* & \dots & a_{1L}^* \\ A_2^* = a_{21}^* & a_{22}^* & \dots & a_{2L}^* \\ & \vdots & & \\ A_r^* = a_{r1}^* & a_{r2}^* & \dots & a_{rL}^*, \end{cases}$$

Example:

$\mathbf{A} = \{\text{apple, paper, pepper}\}$

$$\mathbf{A}^* = \begin{pmatrix} - & \text{a} & \text{p} & \text{p} & \text{l} & \text{e} & - \\ \text{p} & \text{a} & \text{p} & - & - & \text{e} & \text{r} \\ \text{p} & \text{e} & \text{p} & \text{p} & - & \text{e} & \text{r} \end{pmatrix}$$

4.3 Scoring an MSA

In the case of a linear gap penalty, if we assume independence of the different columns of an MSA, then the score $\alpha(\mathbf{A}^*)$ of an MSA \mathbf{A}^* can be defined as a sum of column scores:

$$\alpha(\mathbf{A}^*) = \sum_{i=1}^L s(a_{1i}^*, a_{2i}^*, \dots, a_{ri}^*).$$

Here we assume that $s(a_{1i}^*, a_{2i}^*, \dots, a_{ri}^*)$ is a function that returns a score for every combination of r symbols (including the gap symbol).

For pairwise alignments there are three types of columns, containing either zero gaps, or a gap in the first sequence, or a gap in the second sequence. The following table shows the 7 possibilities for three sequences:

$$\begin{array}{ccccccc} a_{1i} & - & a_{1i} & a_{1i} & - & - & a_{1i} \\ a_{2j} & a_{2j} & - & a_{2j} & - & a_{2j} & - \\ a_{3k} & a_{3k} & a_{3k} & - & a_{3k} & - & - \end{array}$$

For r sequences, the number of different column types is

$$\sum_{i=0}^{r-1} \binom{r}{i} = 2^r - 1$$

where i is the number of gaps.

4.3.1 The sum-of-pairs (SP) score

How to define the column score s ? For two protein sequences, s is usually given by a BLOSUM or PAM matrix. For more than two sequences, providing such a matrix is not practical, as the number of possible combinations is too large.

Given an MSA \mathbf{A}^* , consider two sequences A_p^* and A_q^* in the alignment. For two aligned symbols u and v we define:

$$s(u, v) = \begin{cases} \text{match score for } u \text{ and } v, & \text{if } u \text{ and } v \text{ are residues,} \\ -d & \text{if either } u \text{ or } v \text{ is a gap, or} \\ 0 & \text{if both } u \text{ and } v \text{ are gaps.} \end{cases}$$

(Note that $u = -$ and $v = -$ can occur together in a multiple alignment.)

The multiple alignment A^* induces a pairwise alignment on any two of the input sequences A_p and A_q .

Define the score of this (not necessarily optimal) pairwise alignment as

$$s(A_p^*, A_q^*) = \sum_{i=1}^L s(a_{pi}^*, a_{qi}^*).$$

The sum-of-pairs score is obtained by adding up the scores of all such pairs of sequences:

$$S(A_1^*, \dots, A_r^*) = \sum_{1 \leq p < q \leq r} s(A_p^*, A_q^*),$$

with $s(-, -) = 0$.

Definition 4.3.1 (Sum of pairs score) *The sum-of-pairs (SP) score of an alignment is defined as*

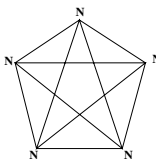
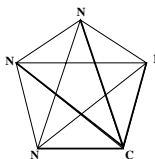
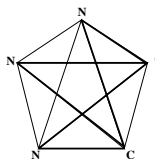
$$\alpha_{SP}(\mathbf{A}^*) = \sum_{1 \leq p < q \leq r} s(A_p^*, A_q^*) = \sum_{i=1}^L s_{SP}(a_{1i}^*, a_{2i}^*, \dots, a_{ri}^*),$$

with

$$s(A_p^*, A_q^*) = \sum_{i=1}^L s(a_{pi}^*, a_{qi}^*) \text{ and } s_{SP}(a_{1i}^*, \dots, a_{ri}^*) = \sum_{1 \leq p < q \leq r} s(a_{pi}^*, a_{qi}^*).$$

This allows us to compute a score to a multiple alignment using a pairwise-scoring matrix such as BLOSUM62.

Let us consider an example.

Multiple alignment:	{		(1)		(2)		(3)		
		Seq. 1	...	N	...	N	...	N	...
		Seq. 2	...	N	...	N	...	N	...
		Seq. 3	...	N	...	N	...	N	...
		Seq. 4	...	N	...	N	...	C	...
		Seq. 5	...	N	...	C	...	C	...
Comparisons:			(1)		(2)		(3)		
									
# comparisons			$\binom{5}{2}=10$		10		10		
N-N pairs:			10		6		3		
N-C pairs:			0		4		6		
C-C pairs:			0		0		1		
BLOSUM62:			60		24		9		

(BLOSUM62 scores: N-N: 6, N-C: -3, C-C: 9)

4.3.2 An undesirable property of the SP score

Consider a position i in an SP-optimal multi-alignment A^* that is *constant*, i.e., has the same residue in all sequences.

What happens when we add a new sequence? If the number of aligned sequences is small, then we would not be too surprised if the new sequence shows a different residue at the previously constant position i .

However, if the number of sequences is large, then we would expect the constant position i to remain constant, if possible.

Unfortunately, the SP score favors the opposite behavior: the more sequences there are in an MSA, the easier it is, relatively speaking, for a differing residue to be placed in an otherwise constant column.

$$\text{Consider } L = \begin{cases} A_1^* = & \dots & \mathbf{x} & \dots \\ A_2^* = & \dots & \mathbf{x} & \dots \\ & \dots & & \\ A_{r-1}^* = & \dots & \mathbf{x} & \dots \\ A_r^* = & \dots & \mathbf{x} & \dots \end{cases} \text{ and } R = \begin{cases} A_1^* = & \dots & \mathbf{x} & \dots \\ A_2^* = & \dots & \mathbf{x} & \dots \\ & \dots & & \\ A_{r-1}^* = & \dots & \mathbf{x} & \dots \\ A_r^* = & \dots & \mathbf{y} & \dots \end{cases}$$

The SP-score of the column in L is

$$s_{SP}(\mathbf{x}^r) = \binom{r}{2} s(\mathbf{x}, \mathbf{x}).$$

The SP-score of the column in R is

$$s_{SP}(\mathbf{x}^{r-1}, y) = \binom{r-1}{2} s(\mathbf{x}, \mathbf{x}) + (r-1)s(x, y).$$

So, the difference between $s_{SP}(\mathbf{x}^r)$ and $s_{SP}(\mathbf{x}^{r-1}, y)$ is:

$$\binom{r}{2} s(\mathbf{x}, \mathbf{x}) - \binom{r-1}{2} s(\mathbf{x}, \mathbf{x}) - (r-1)s(x, y) = (r-1)(s(x, x) - s(x, y)).$$

Therefore, the relative difference is

$$\begin{aligned} \frac{s_{SP}(\mathbf{x}^r) - s_{SP}(\mathbf{x}^{r-1}, y)}{s_{SP}(\mathbf{x}^r)} &= \frac{(r-1)(s(x, x) - s(x, y))}{r(r-1)/2 s(x, x)} \\ &= \frac{2}{r} \left(\frac{s(x, x) - s(x, y)}{s(x, x)} \right), \end{aligned}$$

which *decreases* as the number of sequences r increases!

4.4 Dynamic program for an MSA

Although local alignments are biologically often more relevant, it is easier to discuss global MSA. Dynamic programs developed for pairwise alignment can be modified to multiple alignments. We discuss how to compute a global MSA for three sequences, in the case of a linear gap penalty. Assume we are given:

$$\mathbf{A} = \begin{cases} A_1 = & a_{11} & a_{12} & \dots & a_{1n_1} \\ A_2 = & a_{21} & a_{22} & \dots & a_{2n_2} \\ A_3 = & a_{31} & a_{32} & \dots & a_{3n_3} \end{cases}$$

We proceed by computing the entries of an $(n_1 + 1) \times (n_2 + 1) \times (n_3 + 1)$ -matrix $F(i, j, k)$ recursively. After filling the matrix, the cell $F(n_1, n_2, n_3)$ contains the best score α for a global alignment \mathbf{A}^* . Traceback recovers an optimal alignment.

The main recursion is (remember there are $2^r - 1 = 8 - 1 = 7$ types of columns in this case):

$$F(i, j, k) = \max \begin{cases} F(i-1, j-1, k-1) + s(a_{1i}, a_{2j}, a_{3k}), \\ F(i-1, j-1, k) + s(a_{1i}, a_{2j}, -), \\ F(i-1, j, k-1) + s(a_{1i}, -, a_{3k}), \\ F(i, j-1, k-1) + s(-, a_{2j}, a_{3k}), \\ F(i-1, j, k) + s(a_{1i}, -, -), \\ F(i, j-1, k) + s(-, a_{2j}, -), \\ F(i, j, k-1) + s(-, -, a_{3k}), \end{cases}$$

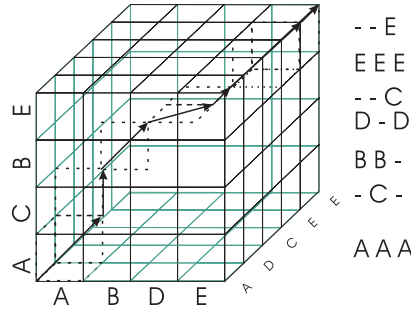
$$\text{for } 1 \leq i \leq n_1, 1 \leq j \leq n_2, 1 \leq k \leq n_3,$$

where $s(a, b, c)$ returns a score for a given column of symbols a, b, c ; for example, $s = s_{SP}$, the sum-of-pairs score.

Example:

$$\mathbf{A} = \begin{cases} A_1 = \text{ABDE} \\ A_2 = \text{ACBE} \\ A_3 = \text{ADCEE} \end{cases} \Rightarrow \mathbf{A}^* = \begin{cases} A_1^* = \text{A} - \text{B} \text{ D} - \text{E} - \\ A_2^* = \text{A} \text{ C} \text{ B} - - \text{E} - \\ A_3^* = \text{A} - - \text{D} \text{ C} \text{ E} \text{ E} \end{cases}$$

Matrix:



Clearly, this algorithm generalizes to r sequences. It has space complexity $O(n^r)$, where n is the sequence length (assuming equal sequence length for all r sequences). Hence, it is only practical for small r and small n .

And how about time complexity? It is $O(n^r \cdot 2^r \cdot r^2)$ when using the SP-score.

(There are n^r cells, 2^r equations per cell, r^2 calculations per equation.)

Theorem 4.4.1 *Computing an MSA with optimal SP-score is NP-complete.*

4.5 Progressive alignment

The most widely used approach to multiple sequence alignment is called *progressive alignment*.

This operates by constructing a series of pairwise alignments, first starting with pairs of sequences and then later also aligning sequences to existing alignments (“profiles”) and then profiles to profiles. Progressive alignment is a heuristic and does not directly optimize any known global scoring function of alignment correctness. However, it is fast and efficient, and often provides reasonable results.

The various implementations differ (1) in the order in which the sequences are aligned, (2) whether during the alignment process a single multiple alignment is generated or several ones, following a tree structure, and (3) which scoring function is used.

The general algorithm for progressive alignments is as follows:

Algorithm 4.5.1 (Progressive alignment)

Input: A set $\mathbf{A} = \{A_1, \dots, A_r\}$ of sequences

Output: A multiple alignment of \mathbf{A} .

begin

Let \mathcal{C} denote the current set of alignment (profiles)

Initialize: $\mathcal{C} = \{\{A_1\}, \{A_2\}, \dots, \{A_r\}\}$.

repeat

Choose two alignments C_p^* and C_q^* in \mathcal{C}

Remove C_p^* and C_q^* from \mathcal{C}

Compute a “super-alignment” C_s^* of C_p^* and C_q^*

```

    Add  $C_s^*$  to  $\mathcal{C}$ 
  until  $|\mathcal{C}| = 1$ 
  return the final alignment contained in  $\mathcal{C}$ 
end

```

4.5.1 Profile alignment

Two profiles can be aligned with each other using dynamic programming.

Suppose we are given two profiles $\mathbf{A}_1 = \{A_1, \dots, A_r\}$ and $\mathbf{A}_2 = \{A_{r+1}, \dots, A_n\}$. Here, we discuss the alignment of profiles in the case of the SP-score and linear gap scores. In this case, we set $s(-, -) = 0$ and $s(-, a) = s(a, -) = -d$ for any letter $a \neq '-'$.

Definition 4.5.2 (Profile alignment) A profile alignment of \mathbf{A}_1 and \mathbf{A}_2 is an MSA

$$A^* = \begin{cases} A_1^* & = & a_{11}^*, & a_{12}^*, & \dots, & a_{1L}^* \\ & \dots & & & & \\ A_r^* & = & a_{r1}^*, & a_{r2}^*, & \dots, & a_{rL}^* \\ & \dots & & & & \\ A_{r+1}^* & = & a_{r+1,1}^*, & a_{r+1,2}^*, & \dots, & a_{r+1,L}^* \\ & \dots & & & & \\ A_n^* & = & a_{n1}^*, & a_{n2}^*, & \dots, & a_{nL}^*, \end{cases}$$

obtained by inserting whole columns of gaps into either \mathbf{A}_1 or \mathbf{A}_2 , without changing the alignment of either of the two profiles.

The SP-score of the profile alignment A^* is:

$$D_{sp}(A^*) = \sum_{1 \leq p < q \leq n} \sum_{i=1}^L s(a_{pi}^*, a_{qi}^*) = \sum_{i=1}^L \sum_{1 \leq p < q \leq n} s(a_{pi}^*, a_{qi}^*) =$$

$$\boxed{\sum_{i=1}^L \sum_{1 \leq p < q \leq r} s(a_{pi}^*, a_{qi}^*)}_{(a)} + \boxed{\sum_{i=1}^L \sum_{r < p < q \leq n} s(a_{pi}^*, a_{qi}^*)}_{(b)} + \boxed{\sum_{i=1}^L \sum_{1 \leq p \leq r < q \leq n} s(a_{pi}^*, a_{qi}^*)}_{(c)}.$$

The sums (a) and (b) are the alignment scores of \mathbf{A}_1^* and \mathbf{A}_2^* .

The third sum (c) consists of all cross terms and can be optimized using standard pairwise alignment, with the modification that columns are scored against columns by adding their pair scores.

Either or both profiles may consist of a single sequence. In the former case, we are aligning a single sequence to a profile and in the latter case, we are simply aligning two sequences.

In the following example, use 1 for match and -1 for mismatch or gap:

$$\begin{aligned} \text{Alignment 1: } & \begin{cases} A_1^* & = & \text{A} & \text{G} & \text{C} & \text{A} & \text{T} \\ A_2^* & = & \text{C} & \text{G} & \text{C} & \text{A} & \text{T} \\ A_3^* & = & \text{C} & \text{G} & \text{A} & \text{T} & - \\ A_4^* & = & \text{A} & \text{G} & \text{A} & \text{T} & - \end{cases} \\ \text{Alignment 2: } & \begin{cases} A_5^* & = & \text{A} & \text{C} & \text{A} & \text{T} \\ A_6^* & = & \text{A} & \text{C} & - & \text{T} \end{cases} \end{aligned}$$

What is the score for each alignment? What is the optimal score for a profile alignment of the two?

Two such profiles are aligned by dynamic programming using a modification of the Needleman-Wunsch or Smith-Waterman algorithm.

	A	G	C	A	T
	C	G	C	A	T
	C	G	A	T	–
	A	G	A	T	–
A	A				
C	C				
–	A				
T	T				

4.5.2 Order of profile alignments matters

Consider the following four amino acid sequences:

$A_1 = \text{ALVK}$, $A_2 = \text{APFK}$, $A_3 = \text{ALFVK}$, $A_4 = \text{APFVK}$.

If we perform the alignments in different orders then we obtain different multiple sequence alignments:

(A_1, A_2) and (A_3, A_4) or (A_1, A_3) and (A_2, A_4) :

ALFK		ALF-K		AL-FK		AL-FK
APFK		APF-K		ALFVK		ALP-K
	→	ALFVK			→	ALFVK
ALFVK		APFVK		ALP-K		APFVK
APFVK				APFVK		

4.5.3 Guide trees

An important part of progressive alignment heuristics is determining the order in which sequences and profiles are aligned.

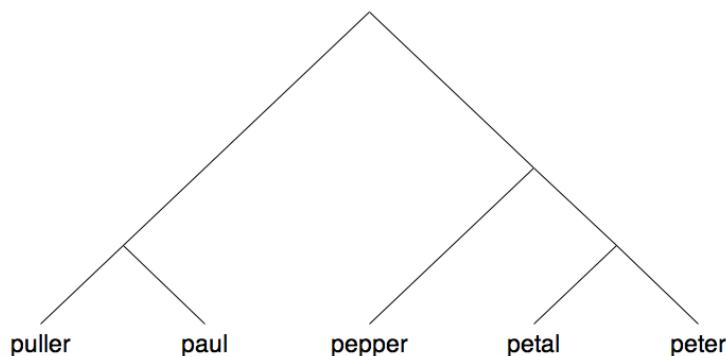
It is a good idea to align the most similar sequences first, and then to add the less similar ones later.

To do this, most progressive algorithms first build a so-called *guide tree*, which is a “rooted tree” whose leaves are labeled by the sequences that are to be aligned.

Sequences are aligned bottom-up along the guide tree, first aligning pairs of sequences, then sequences against profiles (sub-alignments) and then profiles against profiles.

Different algorithms use different methods to compute the guide tree.

Consider the sequences **peter**, **petal**, **paul**, **pepper**, **puller**. Try to align them along the following guide tree:



4.5.4 Feng-Doolittle

The first progressive alignment algorithm was published in 1987 by Feng and Doolittle¹.

Algorithm 4.5.3 (Feng-Doolittle)

1. Calculate all $\binom{r}{2}$ pairwise alignment scores and convert them into distances.
2. Construct a rooted guide tree from the distance matrix using the UPGMA algorithm (described in a later chapter).
3. Build a multiple alignment bottom-up along the guide tree and return the alignment of all sequences that is produced at the root of the tree.

The distance score used by Feng-Doolittle is:

$$D(a, b) = -\log S_{\text{eff}}(a, b) = -\log \frac{S_{\text{obs}}(a, b) - S_{\text{rand}}(a, b)}{S_{\text{max}}(a, b) - S_{\text{rand}}(a, b)},$$

where

- a and b are two sequences,
- $S_{\text{obs}}(a, b)$ is the “observed” similarity score for a and b computed by pairwise alignment of a and b ,
- $S_{\text{max}}(a, b)$ is the maximum possible score, given by $S_{\text{max}}(a, b) = \frac{S(a,a)+S(b,b)}{2}$, and
- $S_{\text{rand}}(a, b)$ is the expected score of an alignment of two *random* sequences of the same length and composition (computed by repeatedly shuffling the letters in a and b).

The sequence-sequence alignments are conducted using the profile alignment approach.

4.5.5 CLUSTALW

CLUSTALW² was for many years the most popular program for computing an MSA. Many more recent programs can be considered improvements of this approach.

Algorithm 4.5.4 (ClustalW progressive alignment)

1. Construct a distance matrix of all $\binom{r}{2}$ pairs by pairwise dynamic programming alignment followed by approximate conversion of similarity scores to evolutionary distances.
2. Construct a guide tree using the Neighbor-Joining tree-building method from the distance matrix.
3. Progressively align sequences at nodes of tree in order of decreasing similarity, using sequence-sequence, sequence-profile and profile-profile alignment.

There are no provable performance guarantees associated with the program. However, it works well in practice and the following features contribute to its accuracy:

- Sequences are weighted to compensate for the defects of the SP score.

¹Feng, D-F & Doolittle, RF. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. J. Mol. Evol. 25:351-360, 1987

²Thompson, J.D., Higgins, D.G. & Gibson, T.J. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. Nucleic Acids Research, 22:4673-4680, 1994.

- The substitution matrix used is chosen based on the similarity expected for the alignment, e.g. BLOSUM80 for closely related sequences and BLOSUM50 for less related ones.
- Position-specific gap-open profile penalties are multiplied by a modifier that is a function of the residues observed at the position (hydrophobic residues give higher gap penalties than hydrophilic or flexible ones.)
- Gap-open penalties are also decreased if the position is spanned by a consecutive stretch of five or more hydrophilic residues.
- Gap-open and gap-extension penalties increase, if there are no gaps in the column, but gaps nearby. (This tries to force gaps to occur in the same places.)
- In the progressive alignment stage, if the score of an alignment is low, then the low scoring alignment may be deferred until later.

The program *T-Coffee* is similar to CLUSTALW, but retains and uses the initial pairwise alignments to produce a better alignment.

4.5.6 Example

We want to align 11 Trypsin and Trypsin inhibitor sequences.

Input: the sequences in a multiple FASTA format (e.g.)

```
>EETI-II
GCPRIILMRCKQSDCLAGCVCGPNGFCGSP
>Ii_Mutant
GCPRIILMRCKQSDCLAGCVCGPNGFCG
>BDTI-II
RGCPRIILMRCKRSDCLAGVCQKNGYCG
>CMeTI-B
VGCPRIILMKCKTDRDCLTGCTCKRNGYCG
>CMTI-IV
HEERVCPRIILMKCKKSDCLAECVLEHGYCG
>CSTI-IIB
MVCPKILMKCKHSDCLLDVCLEDIGYCGVS
>MRTI-I
GICPRILMECKRSDCLAQCVCKRQGYCG
>Trypsin
RICPRIWMECTRSDCMAKCICVAGHCG
>ITRA_MOMCH
RSCPRIWMECTRSDCMAKCICVAGHCG
>MCTI-A
RICPRIWMECKRSDCMAQCICVDGHCG
>LCTI-III
RICPRILMECSSDSDCLAECICLENGFCG
```

First step: pairwise scores

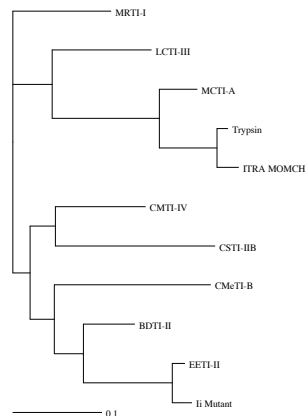
Start of Pairwise alignments

Aligning...

```
Sequences (1:2) Aligned.  Score:  96
Sequences (1:3) Aligned.  Score:  82
Sequences (1:4) Aligned.  Score:  68
Sequences (1:5) Aligned.  Score:  66
Sequences (1:6) Aligned.  Score:  60
Sequences (1:7) Aligned.  Score:  68
Sequences (1:8) Aligned.  Score:  57
```

Sequences (1:9) Aligned. Score: 57
 Sequences (1:10) Aligned. Score: 60
 Sequences (1:11) Aligned. Score: 68
 ...

Second step: the NJ guide tree



Third step: Progressive alignment along the guide tree;

Start of Multiple Alignment

There are 10 groups

Aligning...

Group 1: Sequences: 2 Score:641
 Group 2: Sequences: 3 Score:600
 Group 3: Sequences: 4 Score:571
 Group 4: Sequences: 2 Score:601
 Group 5: Sequences: 6 Score:540
 Group 6: Sequences: 7 Score:561
 Group 7: Sequences: 2 Score:639
 Group 8: Sequences: 3 Score:619
 Group 9: Sequences: 4 Score:560
 Group 10: Sequences: 11 Score:515
 Alignment Score 7716
 CLUSTAL-Alignment file created

Result:

```

          **::*.*.***:  *  *  *  ***
EETI-II  ----GCPRIILMRCKQSDCLAGCVCGPN-GFCGSP  30
Ii_Mutant ----GCPRIILMRCKQSDCLAGCVCGPN-GFCG--  28
BDTI-II  ----RGCPRIILMRCKRSDCLAGCVCKN-GYCG--  29
CMeTI-B  ----VGCPRIILMKCKTDRDCLTGCTCKRN-GYCG--  29
CMTI-IV  --HEERVCPRIILMKCKKSDCLAECVLEH-GYCG--  32
CSTI-IIB ----MVCPIILMKCKHSDCLLDVCLEDIGYCGVS  32
MRTI-I   ----GICPRIILMECKRSDCLAQCVCCKRQ-GYCG--  29
Trypsin  ----RICPRIWMECTRSDCMKACICVA--GHCG--  28
ITRA_MOMCH ----RSCPRIWMECTRSDCMKACICVA--GHCG--  28
MCTI-A   ----RICPRIWMECKRSDCMAQCICVD--GHCG--  28
LCTI-III ----RICPRIILMECSSDSDCLAECICLEN-GFCG--  29
ruler 1.....10.....20.....30.....

```

4.5.7 Run time

The most time-costly part of the ClustalW algorithm is the computation of the initial pairwise alignments:

Number of protein sequences (average length)	200 (412)	400 (468)	600 (462)	800 (454)	1000 (446)
ClustalW (Pentium IV, 3 GHz)					
Overall	194.9	891.9	1818.1	3157.6	4711.6
Pairalign	183.8 (94.4%)	833.1 (93.4%)	1697.0 (93.3%)	2966.6 (94%)	4409.6 (93.6%)
Guided tree	0.07 (0.03%)	0.8 (0.09%)	4.1 (0.2%)	8.0 (0.2%)	16.0 (0.3%)
Malign	11.0 (5.6%)	58.0 (6.5%)	117.0 (6.4%)	183.0 (5.8%)	286.0 (6.1%)

(Source: Oliver et al., Bioinformatics, 21(16):3431-2, 2005)

4.5.8 Improvements

To improve the speed of progressive alignment, more recent methods such as MAFFT³ and Clustal Omega⁴ use fast heuristics to compute a guide tree, for example, based on the comparison of k -mers. The heuristic employed by Clustal Omega runs in $O(n \log n)$ time and was used to align $n = 380,000$ tRNA sequences.

To improve alignment accuracy, Clustal Omega aligns pairs of “Hidden Markov Models” (HMMs) in the progressive alignment stage.

4.6 Summary

- Multiple alignments are alignments of two or more sequences.
- Dynamic programming is impractical for aligning more than two sequences.
- Multiple alignments are scored with the help of pair-wise scoring schemes, e.g. via the sum-of-pairs approach
- Many fast multiple alignment programs are based on the progressive alignment approach.

³Katoh K, and Toh H, Brief Bioinform 2008;9:286a-298

⁴Sievers et al, Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega (2011), Molecular Systems Biology 7(1)