The AMPS Package for Multiple Protein Sequence Alignment

Geoffrey J Barton

Laboratory of Molecular Biophysics

The Rex Richards Building

South Parks Road

Oxford OX1 3QU

Contents

1 Introduction **3**

2 Materials **5**

3 Method **5**

3.1 General 5

3.1.1 Sequence File 7

3.1.2 Pairscore Matrix File 8

3.1.3 MULTALIGN Command File 8

4 Running a Pairwise Sequence Alignment **9**

4.1 Results of a Pairwise Sequence Comparison Run 12

5 Multiple Sequence Alignment **16**

5.1 Single Order Alignment 16

5.2 Tree Alignment 21

6 Notes **26**

6.1 Gap Penalties and Sequence Lengths 26

6.2 Selection of Sequences for Multiple Alignment 27

6.3 Quality of alignment 27

6.4 Time Considerations 28

6.5 File Name Conventions 29

6.6 Single order or tree method? 29

6.7 Aligning Nucleic Acid Sequences 30

6.8 Possible Disasters 30

6.9 VAX/VMS differences 31

7 References **32**

8 Figure Legends **34**

8.1 Figure 1 34

8.2 Figure 2 34

8.3 Figure 3 34

8.4 Figure 4 34

8.5 Figure 5 35

8.6 Figure 6 35

1 Introduction

AMPS (Alignment of Multiple Protein Sequences) is a suite of programmes for the multiple alignment of protein sequences. The programmes build upon the sequence comparison technique introduced in 1970 by Needleman and Wunsch [1]. The Needleman and Wunsch algorithm when supplied with two protein sequences and a matrix of scores for matching each possible pair or amino acids (e.g. the identity matrix, or Dayhoff's mutation data matrix [2]) calculates the best alignment between the two sequences and a score for the alignment. Although the Needleman and Wunsch algorithm may be used with virtually any pair of sequences, extending the method to more than 3 sequences is impractical even with todays fast computers. In order to cope with the alignment of large families of protein sequences, a heuristic approach was therefore developed [3, 4].

The protein sequences to be aligned are first compared pairwise using a conventional Needleman and Wunsch algorithm. Thus, for four sequences $A, B, C\&D$ separate alignments of

$A\&B, A\&C, A\&D, B\&C, B\&D$ and $C\&D$ are made. For each of these alignments, various statistics are gathered (e.g. % identity), and this information is used to group the sequences using a conventional cluster analysis technique. The information gained from the cluster analysis enables a degree of confidence to be assigned to the alignments. Finally, a multiple sequence alignment is obtained either by following the tree produced by cluster analysis, or simply by aligning the sequences in order of their similarity (see below). The result is a multiple sequence alignment that groups the most similar sequences together in the output. In order to assist in the interpretation of the alignment, conservation values [5] may be calculated for each position of the alignment to highlight the positions where the amino acids share similar physico-chemical properties.

AMPS contains additional functions to allow the position of gaps in the alignment to be restricted [6], and to allow specific residue weights to be assigned. AMPS also has features for flexible pattern matching and databank scanning [7], however these aspects will not be discussed in this article.

2 Materials

The AMPS package may be supplied for Digital VAX computers running the VMS operating system, Sun Microsystems SPARC workstations, and Silicon Graphics Iris workstations running Unix. No special terminals or printers are required to use the programmes, though a printer capable of using small fonts is useful if large multiple sequence alignments are to be studied. The package is not available for IBM PC's or Apple computers.

3 Method

3.1 General

A multiple sequence alignment is obtained in three stages using the programmes MULTALIGN and ORDER.

1. The programme MULTALIGN is run in pairwise mode to calculate all pairwise alignments of the sequences and the results are saved to a file. Randomizations may be carried out at this stage to give standard deviation (S.D.) scores for the comparisons and provide an estimate of confidence in the alignments [6, 4]. Normally, only statistics for the

alignments are saved, however, optionally, each pairwise sequence alignment may be written to a file if required.

2. The programme ORDER is run to perform cluster analysis on the sequence scores obtained from running MULTALIGN in stage 1. An *order file* and *tree file* are written out and a simple phylogenetic tree is drawn.

3. MULTALIGN is supplied with the *tree file* and *order file* and run in multiple mode, thus generating a multiple sequence alignment.

In order to make use of the programmes, you must first get to know how to use an editor on the computer. On a VAX/VMS system you might use the editor *EDT* whilst under Unix, the most generally available, though awkward to use editor is called *vi*. Your system manager should be able to help here. In the following description, I will assume you are familiar with the concept of computer files, know how to use the editor on your computer system, and have found out where the AMPS package is located. All the examples that follow assume that you are using a Unix system (e.g. Sun SPARCstation). There are small differences in the way files are defined when using a VAX and these are described in the Notes section.

3.1.1 Sequence File

You must first create a file containing the sequences to be aligned. An example file showing some of the possible formats (i.e. the way in which the information is laid out in the file) is illustrated in Figure 1. The format for each sequence entry has three parts:

1. An identifier code prefixed by the > symbol and of up to nine characters in length.
2. Title line of up to 500 characters in length.
3. The one-letter code amino acid sequence terminated by a * symbol.

All the sequences to be aligned must be put in this format in one file. If you have access to the NBRF-PIR sequence database and the programme PSQ, then the PSQ COPY/NOTEXT command will create files of a suitable format. MULTALIGN will only read the alphabetic characters from the sequence lines. Valid amino acid codes will be read in either upper or lower case, non valid codes (e.g. U) will be read as X (i.e. the unknown amino acid type). All other characters are ignored (e.g. %, 3, 7, blank, etc.).

3.1.2 Pairscore Matrix File

Several pairscore matrix files are supplied with the programme. An example is shown in Figure 2. The file consists of a title line (`Mutation Data Matrix (250 PAMs)` in Figure 2), an amino acid index line (`ARNDCQEGHILKMFPSTWYVBZX` in Figure 2), and a square matrix where each element represents the score associated with matching the pair of amino acids indicated in the row and column. For example, the first row of the matrix in Figure 2 gives the scores for matching `A` (Alanine) with each amino acid (e.g. `AA` $= 2$, `AC` $= -2$). The second row gives scores for `R` (Arginine) versus each amino acid, and so on.

You can customise the pairscore matrix if you wish, or generate your own matrices, perhaps for a specific family of proteins. However, this usually gives little advantage over using the standard Mutation Data Matrix supplied with the programme.

3.1.3 MULTALIGN Command File

As the name suggests, the *command file* contains instructions that tell the MULTALIGN programme what to do, which sequence file to work with, and where to write the results. The command file consists of a number of lines, each of which consists of a keyword followed by an equals sign '=' which may in turn be followed by one or more arguments. The commands may be given in upper or lower case or mixed case. For example, the commands:

```
OUTPUT_FILE=junk.out
output_file=junk.out
OuTpUt_FiLe=junk.out
```

are all equivalent.

Command files must always start with the `OUTPUT_FILE=` command, optionally followed by the `MODE=` command. Most other commands may be specified in any order.

There are many possible combinations of commands these will be illustrated with reference to the alignment of the eight immunoglobulin domains shown in Figure 1.

4 Running a Pairwise Sequence Alignment

If we assume that the immunoglobulin domains shown in Figure 1 are stored in a file called `ig.seq`, the following command file will instruct

MULTALIGN to perform all possible pairwise alignments.

Command File: `ig_pairs.com`

```
output_file=ig_pairs.out

mode=pairwise

matrix_file=md.mat

pairwise_random=100

gap_penalty=8.0

constant = 8

seq_file=ig.seq
```

`output_file=ig_pairs.out` tells MULTALIGN to write the results to the file `ig_pairs.out`. A command of this type must ALWAYS be the first. (Note: on VAX/VMS systems the output file should NOT be set to the log file name.)

`mode=pairwise` specifies that pairwise alignments are to be performed. If this command is not included, the program defaults to `mode=multiple`.

`matrix_file=md.mat` defines the matrix file to be used in the comparisons. This file is the Dayhoff mutation data matrix or similar file containing values for the substitution of each possible amino acid pair (See Figure 2 for an example).

`pairwise_random=100` specifies that 100 randomizations of each sequence pair are to be performed in order to provide statistical S.D. (Standard Deviation) scores.

`gap_penalty=8.0` defines the penalty for inserting a gap (amino acids aligned with one or more blanks) in the alignment to the value of 8.0. Experience shows that this value is adequate for virtually all protein comparisons (but see Notes).

`constant = 8` A constant added to the matrix_file to remove all negative elements. The reason for this is a little obscure, however, a value of 8 is suitable for the matrix in Figure 2.

`seq_file=ig.seq` specifies the file containing the sequences to be aligned pairwise.

This command file causes all pairwise comparisons to be performed on the sequences in the file `ig.seq`. In other words, for the 8 sequences sequence 1 is aligned with 2, 1 with 3 and so on. For N sequences there are

$N(N-1)/2$ comparisons performed. In order to execute the command file on a Unix system, simply type:

```
multalign < ig_pairs.com
```

For each sequence pair (eg. 1 and 2), a full Needleman and Wunsch sequence comparison is performed. Then the sequences are shuffled and recompared (in this case 100 times) in order to find the expected distribution of scores that would be obtained if the sequences were unrelated but have the same length and composition as 1 and 2. Various statistics on the comparisons are then output to the specified `output_file` (`ig_pairs.out`).

4.1 Results of a Pairwise Sequence Comparison Run

Figure 3 shows the output file (`ig_pairs.out`) that results from running MULTALIGN with the command file `ig_pairs.com`. This file contains a great deal of useful information which is described in this section. However, if your principal aim is to generate a multiple sequence alignment, then skip to the section 'Multiple Sequence Alignment'.

The file firstly shows the programme name, its original source and limitations of use. Next come the programme limits. The important values to note are the `Maximum allowed sequence length` and `Maximum number of sequences`. The version of MULTALIGN shown will handle up to 500 sequences of 500 amino acids in length, but a version able to cope with 250 sequences of 1200 length is also distributed. Various information regarding the parameters selected in the `command file` are then shown before listing information about the sequences present in the `seq_file`.

`Index` is a number used to refer to the sequence. Here, it is simply the order in the sequence data file (Figure 1).

`Identifier` shows the identifier from the sequence data file.

`Length` gives the number of amino acids in the sequence.

`Name` shows the title information from each sequence entry.

`File order`, `Score` and `with` should all be zero, since these columns are only relevant after multiple sequence alignment (see below).

There follows a list of the statistics obtained from all the pairwise sequence comparisons. There are 16 columns headed `I, J, ILEN, JLEN, MATCH, NGAPS, NALIG, NIDENT, %IDENT, NAS, NASAL, NRANS, RMEAN, STDEV, SCORE` and blank.

`I` & `J` show the index numbers of the sequence compared in the row. For example, row one shows the comparison of sequence 1 (1fb4hv) with 2 (1fb4hc).

ILEN & JLEN give the lengths of the sequences compared.

MATCH is the raw match score obtained for the comparison of the two sequences. This value is not normalized, i.e. it increases as the length of the sequences compared increases.

NGAPS is the number of internal gaps in the alignment (overhangs at the ends are not counted).

NALIG shows the number of positions at which two amino acids are aligned (i.e. not aligned with a gap).

NIDENT gives the number of positions at which identical amino acids are aligned.

%IDENT shows the percentage identity (i.e. %IDENT/NALIG).

NAS gives a Normalised Alignment score - The match score divided by the number of aligned positions * 100. (MATCH/NALIG)×100.

NASAL shows an alternative Normalised Alignment score - The match score minus the number of gaps times the gap penalty all divided into the number of aligned positions. (MATCH/((MATCH-NGAPS)*gap_penalty))× 100. NRANS records the number of randomizations performed for this pair.

RMEAN shows the mean score for the randomizations.

STDEV gives the standard deviation of the random scores.

SCORE shows the S.D. score for the alignment. Given by the mean random score minus the match score all all divided by the standard deviation.

(RMEAN - MATCH)/STDEV

Column 16 simply numbers each pairwise comparison for reference purposes.

This output file is in the correct format for input by the program ORDER to perform cluster analysis prior to multiple sequence alignment. However, the actual alignments that give the statistics shown have not been output. If these are required, you must include the command line(s):

`print_horizontal=` (for horizontal format) and/or `print_vertical=` (for vertical format). The alignment output formats are the same as those described in the following sections on multiple sequence alignment. Note however, that a file containing alignments cannot be read directly by ORDER. (the alignments would first have to be deleted using a text editor). Alternatively, the command: `pairwise_align_file=name`, where name is a file name that does not already exist on the computer, will instruct MULTALIGN to write the sequence alignments to the specified file.

5 Multiple Sequence Alignment

The results of the pairwise sequence comparison run described in the previous section are used to guide the multiple sequence alignment. Two possible strategies for multiple alignment are supported by AMPS, *single order* and *tree*. Both methods require the programme ORDER to be run on the results of the pairwise analysis in order to generate an `order_file` and for *tree* alignment, a `tree_file`.

5.1 Single Order Alignment

To run the programme ORDER, simply type `order` and follow the steps shown below. What you type is shown in **bold**, **return** means press the Return key.

order

```
---------------------
Program O R D E R
---------------------
Processes MULTALIGN pairwise output
Author:  Geoff Barton
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

This program is supplied for use on a single machine

and should not be copied or redistributed

without prior written consent from the author.

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Maximum Number of sequences allowed:  250

Enter pairwise filename:  ig_pairs.out

Does the input file have timings?  (Def = N) :return

-------------------------------

number of lines read in = 28

-------------------------------

ENTER FILENAME FOR COMPARISONS TO EXCLUDE [none]return

Write out prolog format file [N]: return

Options:  NGAPS(1),PIDENT(2),NAS(3),NASAL(4),RSCORE(5) ?

Enter choice :5

---------------------------

Number items compared 8

---------------------------

Number of identifiers 8
```

```
---------------------------

Produce an order file?  [Y] Y

Enter filename for order ig_pairs.ord

Order file written

Re-analyze the data?  [Y] N

Program ORDER ends
```

The terminal session shown, uses the S.D. scores calculated from randomization to create an order_file. If S.D. scores have not been calculated, then NAS, NASAL, or PIDENT scores can be used instead.

Having generated the order_file, MULTALIGN may be run to generate a multiple sequence alignment by supplying the programme with the following command file:

Command File: `ig_mult.com`

```
output_file=ig_mult.out

mode=multiple

matrix_file=md.mat

gap_penalty=8.0
```

```
constant = 8

seq_file=ig.seq

order_file=ig_pairs.ord

print_vertical=

print_horizontal=

consplot=mz
```

Note the changes from the file `ig_pairs.com`. The `mode` command is now set to `multiple`, the `pairwise_random` command is deleted, and four new commands are added: `order_file=ig_pairs.ord` tells the programme to read the order_file generated by the programme ORDER. `print_vertical=` and `print_horizontal=` instruct MULTALIGN to write out the sequence alignment in both vertical and horizontal formats, whilst `consplot=mz` asks for a conservation profile [5] to be generated for the alignment. MULTALIGN is run using the `ig_mult.com` file by typing:

```
multalign < ig_mult.com
```

Figure 4. illustrates the output of the single order multiple alignment of the eight immunoglobulin domains. Single order alignment

starts by aligning the most similar pair of sequences, then aligns the next most similar sequence to that pair, then the next most similar sequence to the triple, and so on until all 8 sequences are aligned. The `File order`, `Score` and `with` columns show how the order of alignment was determined. For example, the sequence with the Index `3` (1fb4lc) is the 8th sequence in the original sequence file `ig.seq` and shows the greatest similarity of 9.93 S.D. with sequence 2 (1fb4h).

After the programme descriptions, there follow two forms of multiple alignment output. The first is the result of the `print_vertical=` command which prints out the sequences in columns. Alongside each aligned position in this alignment, the `consplot=mz` command has led to the generation of a conservation profile with values between 0 and 1 where 1 equals total identity, and high values show positions with conserved physico-chemical properties (see [5] for details of conservation values). The conservation values are also shown graphically by a simple graph built up of crosses. Below the vertical format alignment, a more conventional horizontal format alignment is shown as the result of the `print_horizontal=` command. The number of residues printed per line can be varied by including the `max_horiz=` command. For example, if 60 residues per line were required, then putting `max_horiz=60` in the MULTALIGN command file would have the desired effect.

5.2 Tree Alignment

Although the single order alignment technique is very effective, a more general approach is to follow the branches of a phylogenetic tree (eg. see Feng and Doolittle [8]). This method has the advantage of grouping the sequences into clusters so that the most similar sequences are close together in the alignment. The AMPS package provides a straightforward procedure for generating alignments of this type.

As before, we must first run the ORDER programme:

order

```
---------------------

Program O R D E R

---------------------

Processes MULTALIGN pairwise output

Author:  Geoff Barton

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
This program is supplied for use on a single machine

and should not be copied or redistributed

without prior written consent from the author.

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Maximum Number of sequences allowed:  250

Enter pairwise filename:  ig_pairs.out

Does the input file have timings?  (Def = N) : return

-------------------------------

number of lines read in = 28

-------------------------------

ENTER FILENAME FOR COMPARISONS TO EXCLUDE [none]return

Write out prolog format file [N]: return

Options:  NGAPS(1),PIDENT(2),NAS(3),NASAL(4),RSCORE(5) ?

Enter choice :5

---------------------------

Number items compared 8

---------------------------

Number of identifiers 8

---------------------------
```

```
Produce an order file?  [Y] N

Perform cluster analysis?  [Y] return

Save full cluster details?  [Y] return

Enter file to save cluster order:  ig_pairs.tord

Enter file to save tree details:  ig_pairs.tree

7 CLUSTERS GENERATED

8 ELEMENTS CONSIDERED

Enter plot filename ig_pairs.plot

Please enter the axis title :S. D. Score

Please enter the plot title :Ig Domains

Enter page width required :80

-------------------------

Cluster analysis complete

-------------------------

Re-analyze the data?  [Y] n

Program ORDER ends
```

This time we answer NO to the option to produce an order file and YES to cluster analysis. The file `ig_pairs.tord` will be similar, but not usually identical to the `ig_pairs.ord` file generated in single order mode.

This file tells MULTALIGN the order in which to output the sequences once they are aligned. The new file required by MULTALIGN to perform tree alignments is `ig_pairs.tree`. In a addition, a plot file is generated where the page width should be given as the largest number of characters that your printer will comfortably plot on one line.

The plot file `ig_pairs.plot` is illustrated in Figure 5. The right hand side of the file shows the numbers and identifying codes of the sequences aligned. From each sequence identifier there extends a horizontal line which terminates in a cross alongside a similar cross at some point on a line from another sequence identifier. This point shows the level at which the two sequences are similar as indicated by the numbers running across the top of the figure. For example, `3fabhc` and `1fb4hc` are the most similar pair and are joined at a score of 30.55 S.D. whilst `3fablc 1fb4lc` are the next most similar pair (26.19 S.D.). These four sequences are then joined into a single cluster at 8.79 S.D.. With a little practice, the eye can readily pick out the major groupings within a set of sequences using plots of this kind. In Figure 5, sequences 2, 4, 6 & 8 form one clear cluster (the Ig constant domains) and sequences 1,3,5 & 7 form another cluster (the Ig variable domains). The two clusters show only limited similarity to each other.

A command file to perform tree multiple alignment follows:

Command File: `ig_mult_tree.com`

```
output_file=ig_mult_tree.out

mode=multiple

matrix_file=md.mat

gap_penalty=8.0

constant = 8

seq_file=ig.seq

order_file=ig_pairs.tord

tree_file=ig_pairs.tree

print_vertical=

print_horizontal=

consplot=mz
```

The only difference between this file and the single order command file ig_mult.com is the addition of the tree_file= command, and the selection of ig_pairs.tord order file.

The result of applying *tree* alignment to the Ig sequences is shown in Figure 6. The principal differences between this alignment and that shown in Figure 4 are the order in which the sequences are presented, and the details of the alignment between the Variable domains and Constant domains.

6 Notes

6.1 Gap Penalties and Sequence Lengths

The gap penalty of 8 used in the example is a good starting point. It should be increased (double or treble) if sequences of very different length are being compared, since low gap-penalties can lead to an unrealistic 'smearing' of short sequences over the length of a longer sequence. In general this algorithm works best for sequences of about the same length that are expected to be similar over most of their length. It is therefore best to delete pro-sequences, or extra domains prior to multiple alignment if these are only present in some of the sequences to be aligned.

6.2 Selection of Sequences for Multiple Alignment

Plots such as Figure 5 can suggests which groups of sequences may be reliably aligned. For example, Barton and Sternberg [6, 4] suggested that alignments giving pairwise S.D. scores > 6.0 would be correct within regions of core secondary structure. On Figure 5 this would indicate that sequences 2,4,6 & 8 will be reliably aligned, similarly 1,3,5, & 7, but the complete alignment of all eight Ig domains is likely to be subjet to errors. Such consideration of possible errors in alignment is vital when assessing the biological significance of any similarities found.

6.3 Quality of alignment

If sequences that cluster at > 6.0 S.D. are multiply aligned by AMPS, then the alignment is likely to be correct within the conserved core of the protein. However, it is important to remember that outside these regions, a sequence alignment may have no meaning at the residue-by-residue level. For example, a surface loop on a protein may adopt completely different conformations in two otherwise similar proteins. Clearly, any inference from a sequence alignment of such variable regions will by very suspect. As a rule of thumb it is best to treat parts of the multiple alignment that contain gaps with caution, whilst using the conservation profile as a guide to the most highly conserved, and hence structurally and functionally important parts of the sequences.

6.4 Time Considerations

When a large number of sequences are to be aligned, performing randomizations for all pairs of sequences can be very time consuming. A good guide to the relative similarity between the sequences is given by the normalised alignment scores `NAS` and `NASAL`. These do not require randomization in order to calculate and so considerably speed the process. Simply delete the line `pairwise_random=100` from the command file to avoid performing the randomizations. When running ORDER, you should then select option 3 rather than 5.

6.5 File Name Conventions

To avoid confusion between files and different projects, you should adopt the following conventions for naming the various files used for alignment. For example, if we were to start with a sequence file called PROTEIN.seq we would use the word PROTEIN to identify the other files as follows:

Sequence file	PROTEIN.seq
Pairwise alignment command file	PROTEIN_pairs.com
Result of pairwise comparisons	PROTEIN_pairs.out
Single order file	PROTEIN_pairs.ord
Tree order file	PROTEIN_pairs.tord
Tree file	PROTEIN_pairs.tree
Multiple alignment command file	PROTEIN_mult.com
Plot file (dendrogram)	PROTEIN_pairs.plot
Multiple alignment output	PROTEIN_mult.out

6.6 Single order or tree method?

In general, opt for the tree alignment method as this groups the sequences more closely by similarity. However, this approach is slower than the single order technique and if the overall similarity between the sequences is very high it, offers no advantage.

6.7 Aligning Nucleic Acid Sequences

Although developed for protein sequences, AMPS can also align DNA/RNA sequences. A special DNA matrix is supplied for this purpose.

6.8 Possible Disasters

A common mistake is to leave blank lines in the command file for MULTALIGN, or forget the '=' sign in a command. MULTALIGN will write an error message to the output file and stop if this occurs.

Another common mistake is to forget to put either the `print_vertical=` or `print_horizontal=` commands in the multiple alignment command file. The programme will then happily compute the alignment, but not write it out!

For alignments that are expected to take a long time, it is always best to check that all the parameters have been correctly read, by interrupting the programme after a few minutes, examining the output_file so far, then re-running the programme for the full period.

A more serious error is to attempt to align sequences that are close to the maximum length limits of the programme. Sequences that are too long on input are ignored, but if during alignment, the alignment grows to a length greater than that allowed, an unexpected programme crash may result.

6.9 VAX/VMS differences

For VAX/VMS systems, the example command files shown in this article need two extra lines at the top:

```
$set def directory
$multalign
```

where, `directory` is the name of the directory in which you are working. The `output_file` should not be set to the same name as the VMS log file. The `output_file` will overwrite any file of the same name present in the working directory. This is unlike normal VAX/VMS practice where old versions of a file are retained, so if you want to keep an old version, remember to re-name it before running MULTALIGN.

7 References

References

[1] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48:443–453, 1970.

[2] M. O. Dayhoff, R. M. Schwartz, and B. C. Orcutt. A model of evolutionary change in proteins. matrices for detecting distant relationships. In M. O. Dayhoff, editor, *Atlas of protein sequence and structure*, volume 5, pages 345–358. National biomedical research foundation, Washington DC, 1978.

[3] G. J. Barton. Protein multiple sequence alignment and flexible pattern matching. *Methods Enzymol.*, 183:403–428, 1990.

[4] G. J. Barton and M. J. E. Sternberg. A strategy for the rapid multiple alignment of protein sequences: Confidence levels from tertiary

structure comparisons. *J. Mol. Biol.*, 198:327–337, 1987.

[5] M. J. J. M. Zvelebil, G. J. Barton, W. R. Taylor, and M. J. E. Sternberg. Prediction of protein secondary structure and active sites using the alignment of homologous sequences. *J. Mol. Biol.*, 195:957–961, 1987.

[6] G. J. Barton and M. J. E. Sternberg. Evaluation and improvements in the automatic alignment of protein sequences. *Protein Eng.*, 1:89–94, 1987.

[7] G. J. Barton and M. J. E. Sternberg. Flexible protein sequence patterns - a sensitive method to detect weak structural similarities. *J. Mol. Biol.*, 212:389–402, 1990.

[8] D. F. Feng and R. F. Doolittle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.*, 25:351–360, 1987.

8 Figure Legends

8.1 Figure 1

Title: Example Sequence Data File

A very flexible format is allowed, but all sequences must be stored in a single file in one letter code.

8.2 Figure 2

Title: Example Pairscore Matrix File

Each element of the matrix gives the score for aligning the corresponding amino acids (e.g. AA = 2).

8.3 Figure 3

Title: Example Output File from running MULTALIGN in pairwise mode.

8.4 Figure 4

Title: Result of Single Order multiple alignment on data from Figure 1.

Note: The programme header has been deleted to simplify the Figure. The vertical format alignment has also been truncated to reduce the size of the Figure.

8.5 Figure 5

Title: Dendrogram or tree for data from Figure 3.

The identifying codes for the more similar sequences are shown clustered together.

8.6 Figure 6

Title: Example Output File from Tree multiple alignment using AMPS.