CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

| | |
|---|---|
| Title | The CLUSTAL_X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools |
| Authors | Thompson, Julie D.;Gibson, Toby J.;Plewniak, Frederica;Jeanmougin, Francois;Higgins, Desmond G. |
| Publication date | 1997 |
| Original Citation | Thompson, J. D., Gibson, T. J., Plewniak, F., Jeanmougin, F. and Higgins, D. G. (1997) 'The CLUSTAL_X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools', Nucleic Acids Research, 25(24), pp. 4876-4882. doi: 10.1093/nar/25.24.4876 |
| Type of publication | Article (peer-reviewed) |
| Link to publisher's version | https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/25.24.4876 - 10.1093/nar/25.24.4876 |
| Rights | © 1997, Oxford University Press |
| Download date | 2025-05-07 04:23:39 |
| Item downloaded from | https://hdl.handle.net/10468/5038 |

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

**4876–4882** *Nucleic Acids Research, 1997, Vol. 25, No. 24* © 1997 Oxford University Press

# The CLUSTAL_X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools

**Julie D. Thompson, Toby J. Gibson[1], Frédéric Plewniak, François Jeanmougin* and Desmond G. Higgins[2]**

Institut de Genetique et de Biologie Moleculaire et Cellulaire (CNRS/INSERM/ULP), BP 163, 67404 Illkirch Cedex, France, [1]European Molecular Biology Laboratory, Postfach 10.2209, 69012 Heidelberg, Germany and [3]Department of Biochemistry, University College, Cork, Ireland

Received September 24, 1997; Revised and Accepted October 28, 1997

## ABSTRACT

**CLUSTAL X is a new windows interface for the widely-used progressive multiple sequence alignment program CLUSTAL W. The new system is easy to use, providing an integrated system for performing multiple sequence and profile alignments and analysing the results. CLUSTAL X displays the sequence alignment in a window on the screen. A versatile sequence colouring scheme allows the user to highlight conserved features in the alignment. Pull-down menus provide all the options required for traditional multiple sequence and profile alignment. New features include: the ability to cut-and-paste sequences to change the order of the alignment, selection of a subset of the sequences to be realigned, and selection of a sub-range of the alignment to be realigned and inserted back into the original alignment. Alignment quality analysis can be performed and low-scoring segments or exceptional residues can be highlighted. Quality analysis and realignment of selected residue ranges provide the user with a powerful tool to improve and refine difficult alignments and to trap errors in input sequences. CLUSTAL X has been compiled on SUN Solaris, IRIX5.3 on Silicon Graphics, Digital UNIX on DECstations, Microsoft Windows (32 bit) for PCs, Linux ELF for x86 PCs, and Macintosh PowerMac.**

## INTRODUCTION

The most widely used method in molecular biology to align sets of nucleotide or amino acid sequences, is to build up a multiple alignment progressively (1–2). The most closely related groups of sequences are aligned first and then these groups are gradually aligned together, keeping the early alignments fixed. This approach works well when the sequences are sufficiently closely related. However, a globally optimal solution (or a biologically significant one) cannot be guaranteed. In more difficult cases, where many sequences have <30% residue identity, this automatic method becomes less reliable. Any misaligned regions introduced in previous stages of the progressive alignment are not corrected later as new information from other sequences is added. In such cases, the automatic alignments need to be refined, either manually or automatically.

Numerous sequence editors have been developed which allow the user to display and manually make or modify an alignment (eg. 3–7). These programs are useful for making small refinements to an alignment, but the totally manual alignment of large numbers of sequences is not feasible. Manual alignment is also highly subjective hence is at least as likely as the automatic alignment process to result in errors in the alignment.

The CLUSTAL X interface has been written to provide a single environment in which the user can perform multiple alignments, view the results and, if necessary, refine and improve the alignment. Tools for alignment quality analysis have been developed which allow the user to highlight low-scoring regions in the alignment. Options are available for automatically correcting these low-scoring regions by realigning a misaligned sequence or a selected region of an alignment.

In earlier CLUSTAL programs (8–11), nested text menus provided all the options to do multiple sequence/profile alignments and simple phylogenetic tree generation. The output alignments were written to file for display, printing or further manipulation. With these simple menus, the CLUSTAL programs could be highly portable, and run on essentially all computers. Portability has been a major factor in the widespread usage of the CLUSTAL series for sequence alignment. On the other hand, much more attractive and powerful user interfaces can be built using non-portable windows systems.

The NCBI Software Development Toolkit (Version 1.8, National Center for Biotechnology Information, Bethesda, MD) provides one solution to the windows portability problem. It interfaces between the application code and various host windowing systems including the X Window System, Macintosh windows and Microsoft Windows. We have made use of the toolkit to provide a portable windows interface, termed CLUSTAL X.

*To whom correspondence should be addressed. Tel: +33 388 65 32 71; Fax: +33 388 65 32 01; Email: jeanmougin@igbmc.u-strasbg.fr

CLUSTAL X is a new graphical interface to the CLUSTAL W program which displays the sequence alignment in a window on the screen, allowing the user to move easily between different parts of the alignment. Pull-down menus provide all the options familiar to users of the text-menu-driven CLUSTAL W, plus several new features. A versatile, configurable, colouring system is used to highlight conserved residue features in the alignment. Options to mark suspect regions and realign selected residue ranges give the user more information and control over the alignment process and allow difficult alignments to be gradually built and refined.

## MATERIALS AND METHODS

We make use of the NCBI VIBRANT (Virtual Interface for Biological Research and Technology) development library which acts as an interface between the application code of CLUSTAL X and the host windowing system. The NCBI libraries are linked to the CLUSTAL X code providing mechanisms for displaying windows, menus, buttons etc. In this way, the CLUSTAL X code remains independent of the underlying operating system and computer. The CLUSTAL X code is written in ANSI C, and should be portable to any machine capable of supporting the NCBI Vibrant toolkit.

CLUSTAL X is available for a number of platforms including SUN Solaris, IRIX 5.3 on Silicon Graphics, Digital UNIX on DECstations, Microsoft Windows (32 bit) for PCs, Linux ELF for x86 PCs, and Macintosh PowerMac. The source code is provided for anyone wishing to port to any other platform supported by the Vibrant project.

The source code for CLUSTAL X and several executable versions for different machines are freely available by anonymous ftp to ftp-igbmc.u-strasbg.fr. Hypertext documentation can be viewed at www-igbmc.u-strasbg.fr/BioInfo/clustalx/. The NCBI Vibrant Toolkit is available by anonymous ftp from ncbi.nlm.nih.gov.

### Installation

The CLUSTAL X program is easily installed by copying the executable file to a system directory which can be seen by all users. Several parameter files (named *.par), and an on-line help text file (clustalx.hlp for MS Windows, otherwise clustalx_help) are also required by CLUSTAL X. These files should be copied to one of the following directories: (i) the user's current directory, (ii) the user's home directory, (iii) any of the directories specified by the PATH environment variable.

## RESULTS

### Algorithms checking alignment quality

Three methods of alignment quality analysis are implemented. (i) A 'quality' score is calculated for each column in the alignment, which depends on the amino acid variability in the column. A high score indicates a highly conserved column, a low score indicating a less well-conserved position. The scores are automatically plotted in the window display (Fig. 1). (ii) The residues which get exceptionally low scores in the above quality calculation can be highlighted in the alignment display (Fig. 1). (iii) Low-scoring segments in each sequence of the alignment can be highlighted. These are found by summing negative scores in the profile built from the sequence alignment (Figs 1 and 2).

Highlighted residues may be expected to occur at a moderate frequency in all the sequences because of their steady divergence due to the natural processes of evolution, although the most divergent sequences are likely to have the most outliers. However, the highlighted residues are especially useful in pointing to sequence misalignments. These can arise due to various reasons: (i) partial or total misalignments caused by a failure in the alignment algorithm, (ii) partial or total misalignments because at least one of the sequences in the given set is partly or completely unrelated to the other sequences, (iii) frameshift translation errors in a protein sequence causing local mismatched regions to be heavily highlighted (see Discussion for more details).

Occasionally, highlighted residues may point to regions of some biological significance. This might happen for example if a protein alignment contains a sequence which has acquired new functions relative to the main sequence set.

*Quality scores.* Suppose we have an alignment of M sequences of length N. Then, the alignment can be written as:

$$\begin{array}{lllll} A_{1,1} & A_{1,2} & A_{1,3} & \cdots\cdots\cdots & A_{1,N} \\ A_{2,1} & A_{2,2} & A_{2,3} & \cdots\cdots\cdots & A_{2,N} \\ \vdots & & & & \\ A_{M,1} & A_{M,2} & A_{M,3} & \cdots\cdots\cdots & A_{M,N} \end{array}$$

Suppose we also define a residue comparison matrix C of size R*R, where R is the number of residues. C(a,b) is the score for aligning residue a with residue b (all scores in this matrix are positive).

The problem is to calculate a score for the conservation of the jth position in the alignment. Vingron and Sibbald (12) used a geometric analysis based on a continuous sequence space in order to compare sequence weighting methods. The method defines an N-dimensional space, where N is the length of the alignment. Each sequence can be placed in the space, and the distance between two sequences is defined as the euclidean distance between the sequences in the space. We have applied an analogous approach to each position in the alignment. An R-dimensional space is defined, in which each column of the alignment can be considered. For a specified position j in the alignment, each sequence consists of a single residue which is assigned a point S in the space. For sequence i, position j, the point S is defined as:

$$S = \begin{pmatrix} C(1, A_{ij}) \\ C(2, A_{ij}) \\ \vdots \\ C(R, A_{ij}) \end{pmatrix}$$

We then calculate a consensus value X for the jth position in the alignment. X is defined as:

$$X = \begin{pmatrix} \dfrac{\sum_{i=1}^{R} F_{i,j} * C(i,1)}{M} \\ \dfrac{\sum_{i=1}^{R} F_{i,j} * C(i,2)}{M} \\ \vdots \\ \dfrac{\sum_{i=1}^{R} F_{i,j} * C(i,R)}{M} \end{pmatrix}$$

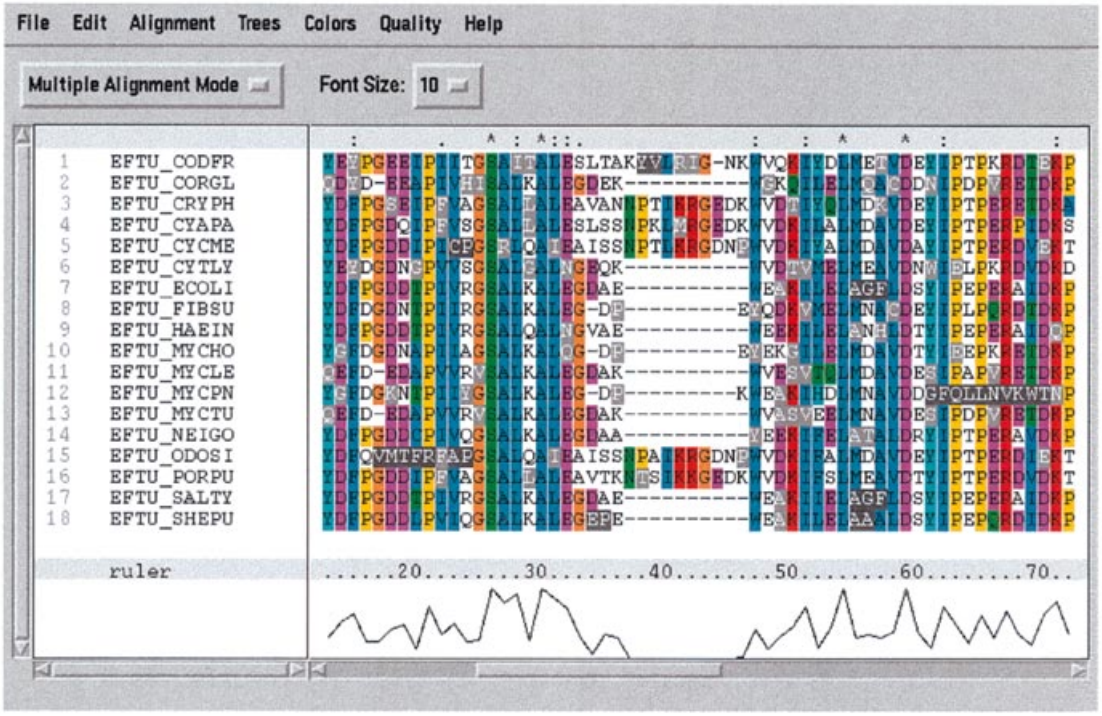


**Figure 1.** The CLUSTAL X window in multiple alignment mode. An alignment of some EFTU proteins is displayed. Low-scoring segments are highlighted using a white character on a black background. Exceptional residues are shown as a white character on a grey background. The quality analysis reveals two anomalously low scoring regions, ruler positions 16–25 in EFTU_ODOSI and 61–71 in EFTU_MYCPN. These were found to be caused by frameshift errors. Two more sequences (EFTU_RICPR and EFTU_SPIPL), not shown here, have 4-residue sequencing errors in this region which CLUSTAL X will also highlight.

where $F_{i,j}$ is the count of residues i at position j in the alignment.

Now, if S is the position of sequence i in the R-dimensional space, we can calculate the distance $D_i$ between each sequence residue i and the consensus position X.

$$D_i = \sqrt{\sum_{r=1}^{R} (X_r - S_r)^2}$$

where $X_r$ is the rth dimension of position X, and $S_r$ is the rth dimension of position S.

We define the quality score for the jth position in the alignment as the mean of the sequence distances $D_i$:

$$Quality\ Score = \frac{\sum_{i=1}^{M} D_i}{M}$$

Finally the scores are normalised by multiplying by the percentage of sequences which have residues (and not gaps) at this position. These scores are used in measure (i) above, as an estimate of the conservation of each alignment column (Fig. 1).

*Exceptional residues.* It would be useful for each column in the alignment to identify those sequences in the above calculations which are found a long way from the consensus point (i.e., which have a large distance $D_i$), thus lowering the quality score for the column. For the jth position in the alignment, we take the set of sequences which have a residue at this position (and not a gap). The distances $D_i$ for this set of sequences are arranged in an array in order of size from smallest to largest. We can find the upper and lower quartiles (the distances lying one-quarter of the way from the top and bottom of the array, respectively) and the inter quartile range (the difference between the two quartiles).

A residue $A_{ij}$ is considered as an exception in measure (ii) above if the sequence distance $D_i$ is greater than (upper quartile + inter quartile range × scaling factor). The scaling factor can be adjusted by the user to select the proportion of residue exceptions that will be highlighted in the alignment display. Exceptional residues in an EFTU alignment are shown in Figure 1.

This calculation runs automatically, in a very short time, each time the screen is updated.

*Low-scoring segments.* Given the above alignment of M sequences of length N and a residue exchange matrix, we can build a profile which is weighted for sequence divergence. Methods for calculating sequence weights are discussed by Henikoff and Henikoff (13). Here we calculate sequence weights directly from a neighbour-joining tree, using the 'branch-proportional' method which corrects for unequal representation by downweighting similar sequences and upweighting divergent ones (14). Each sequence is assigned a weight $W_i$. In the residue comparison matrix C, the scores for common residue substitutions are positive while rarer substitutions are scored negatively. By default, the Gonnet PAM 250 matrix (15) is used, but the user may supply a different matrix e.g. a lower PAM value is appropriate if the sequences are closely related. The profile P has a column of scores for each position in the alignment. The column is of height R and consists of a score

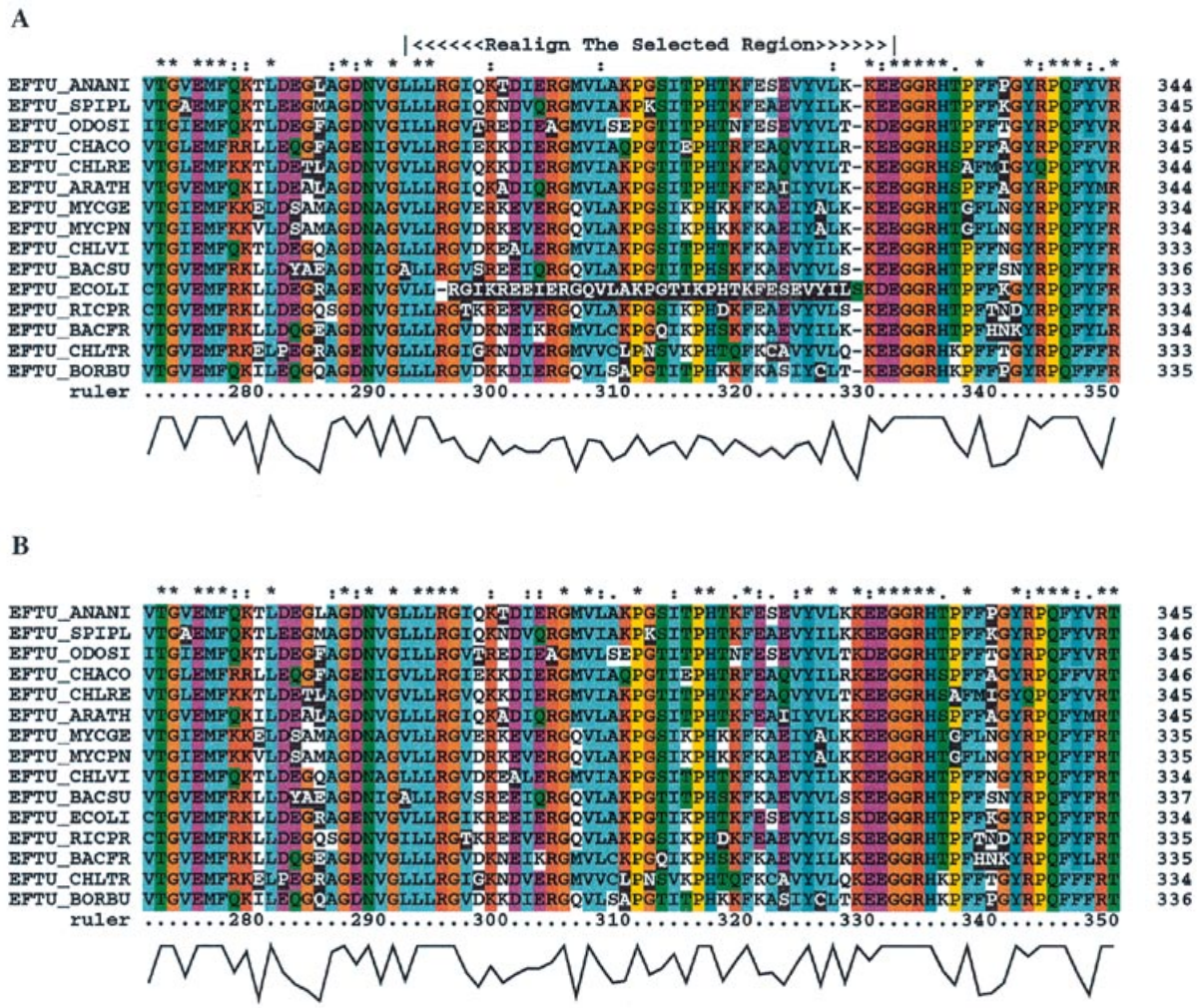


**Figure 2.** Detection and correction of misaligned segments with CLUSTAL X. (**A**) A set of EFTUs, tested for low scoring regions, highlights a part of the EFTU_ECOLI sequence (which we deliberately misaligned by incorrect gap insertion). The range selected to be realigned is marked above the alignment. (**B**) After removal of gaps and realignment of the selected residue range, the sequence EFTU_ECOLI is now correctly aligned, and the erroneous gaps have been removed. The low-scoring segment check, the column conservation indicators above the alignment, and the quality graph below it, all reflect the improvement in the alignment.

for each residue in the matrix C. The profile score for residue r at position j in the alignment is defined as:

$$P(r,j) = \frac{\sum_{i=1}^{M} C(r, A_{ij}) * W_i}{\sum_{i=1}^{M} W_i}$$

For the jth position in the ith sequence the score $S_{ij}$ is defined as:

$$S_{ij} = P(A_{ij}, j)$$

The low-scoring regions in the ith sequence are found by summing the scores $S_{ij}$ along the alignment in both the forward and backward directions. If the sum is found to be positive, it is reset to zero. The forward phase can be described by the following recurrence relations:

$$F_j = \begin{cases} F_{j-1} + S_{ij} & if\ F_{j-1} + Sij < 0 \\ 0 & if\ F_{j-1} + S_{ijh} \geq 0 \\ 0 & if\ \ j = 0 \end{cases}$$

Having found the regions in the sequence which have negative $F_j$ scores, these regions are then refined by removing those positions at the end of each segment which have a positive profile score $S_{ij}$. The $F_j$ scores for these positions are reset to zero.

Similarly, the backward phase can be described as:

$$B_j = \begin{cases} B_{j+1} + S_{ij} & \text{if } B_{j+1} + S_{ij} < 0 \\ 0 & \text{if } B_{j+1} + S_{ij} \geq 0 \\ 0 & \text{if } \ \ j = N + 1 \end{cases}$$

The regions in the sequence which have negative $B_j$ scores are again refined by removing those positions at the beginning of each segment which have a positive profile score $S_{ij}$. The $B_j$ scores for these positions are reset to zero.

The calculation is repeated for each sequence compared with a profile for all aligned sequences, except itself. The low-scoring segments, defined as those positions for which both $F_j$ and $B_j$ are negative, are then highlighted in the display (Figs 1 and 2).

The low-scoring segment calculation is done when the user selects the 'Calculate low scoring segments' option. It takes only a few seconds to perform unless the alignment is very large, making it a practical tool for interactive use.

### Implementation

CLUSTAL X displays a window on the screen, including a set of pull-down menus. On-line help is available. The exact format of the screen will depend on the host computer and the operating system. The user may select one of two modes: (i) multiple alignment mode which displays a single display area for multiple sequence alignment, or (ii) profile alignment mode which has two display areas, allowing the user to use previously aligned sequences for alignment. Figure 1 shows a CLUSTAL X window in multiple alignment mode.

Alignments or individual sequences can be loaded into the display areas displayed on the screen using the menu options. Scroll bars allow easy movement to different parts of the alignment. Extra lines are added to the sequence data displaying a ruler, an indicator of alignment conservation, plus any secondary structure data which was found in the input alignment file.

The order of the sequences in the display can be changed by clicking on the sequence names, and selecting the cut and paste options from the menus. In profile alignment mode, these options also allow the user to move sequences from one profile to the other.

The sequences can be saved at any time to an alignment file in one of a number of file formats. The sequence display can also be saved in a colour postscript file for printing on a postscript printer.

*Colouring the alignment display.* The sequences are automatically coloured to highlight conserved regions of the alignment. The colours used, and the specification of the conservation of residues can be configured by the user. The 'rules' governing the colouring of residues are read from a colour parameter file, which can be loaded at any time. Two types of colouring 'rules' are defined. (i) A residue can be assigned a specific colour regardless of its position in the alignment. In this case, all occurrences of the residue will be coloured in the alignment display. (ii) A residue can be assigned different colours depending on the consensus of the alignment at each position.

In this way, for example, conserved hydrophobic or hydrophilic positions in the alignment can be highlighted (Fig. 1).

*Realigning divergent regions.* In difficult cases, with a family of highly divergent sequences, it is possible that misalignments are introduced during the multiple alignment process. CLUSTAL X provides two simple mechanisms for realigning the most divergent regions. (i) Misaligned sequences may be selected by clicking on the sequence names. A single menu option then removes these sequences from the alignment set and realigns them to the remaining sequences. (ii) The second option allows the user to specify a range of the alignment to be realigned. In this case, the selected sub-range of the alignment is removed and multiply aligned using the standard progressive multiple sequence alignment method. The sub-range is then fitted back into the full alignment.

Using these two options, the original multiple sequence alignment may be iteratively improved and refined.

## DISCUSSION

Ideally, methods for multiple sequence alignment should guarantee to find the biologically correct alignment for a set of sequences. In practice, this is difficult to achieve. Firstly, it is difficult to define an optimal alignment between divergent nucleotide or protein sequences, even given tertiary structural information. Secondly, methods that find an optimal multiple sequence alignment have been impractical to implement, mainly due to their computational cost. As computer performance improves, methods which iterate toward an optimal alignment are likely to become useful (16,17). Meanwhile the heuristical approach of progressive alignment is most often used, as the algorithm is reasonably fast and minimises error in alignments of moderate difficulty. However, because the full information in the sequence set is not used to align each sequence, it can be possible to see one or more misaligned sequence segments in the output alignment. In such cases, the sequences would be expected to align correctly if the full information was used, or if alignment parameters such as gap penalties were adjusted.

When we developed CLUSTAL W, we gave the user the ability to iterate the alignment process by realigning an alignment, or by profile aligning sequences to an alignment. In this way, the user could choose to iterate the alignment process, thereby overcoming some of the defects of progressive alignment. With CLUSTAL X, we have taken this capability further, by building in algorithms to target the problem regions of an alignment and letting the user realign solely the suspect residue ranges. Using these tools, high quality alignments of divergent sequence sets are produced more quickly and with greater confidence than has previously been possible by progressive alignment.

Many programs have been developed which allow, to a greater or lesser degree, manual intervention in the automatic alignment process. For example, SOMAP (18) was designed to run under the DEC VMS operating system. The program allows the user to manually build up a multiple sequence alignment. It can accept automatic alignments created by the original CLUSTAL program (8) to provide a starting-point for the manual editing process. SEAVIEW (7) is a UNIX X Window-based multiple sequence alignment editor which is interfaced to the CLUSTAL W program. SEQPUP (Don Gilbert, Biology Department, Indiana University, Bloomington, IN 47405) is a sequence editor and analysis program which can launch external applications such as CLUSTAL W to perform sequence alignment. SEQLAB [Wisconsin Package Version 9.0, Genetics Computer Group (GCG), Madison, WI] is a graphical user interface based on the OSF/Motif windowing system. It displays sequence alignments on the screen and includes powerful sequence editing facilities. The PILEUP program is interfaced to the SEQLAB editor to perform automatic multiple sequence alignments. Numerous Mac and PC alignment editors have also been developed. Most of these editors will accept alignment output from CLUSTAL programs. However, using CLUSTAL X, the amount of time spent editing alignments by hand should be minimised, while a hand-edited alignment can itself be returned for error checking.

CLUSTAL X is not confined to either VMS or UNIX work-stations but also runs on Macintosh and PC computers. The program provides a flexible approach to the problem of the multiple alignment of large numbers of sequences. The methods used can be applied equally well to both nucleotide and amino acid sequences. An initial automatic alignment using the traditional

progressive, pairwise approach provides a good starting point for further refinement. The alignments are displayed on the screen, and the user can move around easily between different parts of the alignment. A versatile residue colouring scheme based on the conservation of each position in the alignment automatically highlights conserved or special features.

## Alignment analysis and error detection

Tools for alignment quality analysis have been developed and incorporated into the package. A 'quality' estimate for each position in the alignment is plotted on the screen (Fig. 1). Highly conserved positions in the alignment will get a high 'quality' score, whilst either low conservation, or exceptional residues at a partially conserved position, will lower the score for the column. The exceptional residues, which may be due to misalignment of the sequences or simply divergence can be highlighted in the alignment (Fig. 1). Sometimes these may be of biological interest, although most divergence is due to neutral evolutionary processes.

Several methods for calculating the conservation of an alignment column have been developed. Zvelebil *et al*. (19) used physico-chemical properties of amino acids to quantify the conservation of a position in an alignment, in order to predict protein secondary structure. Smith and Smith (20) define the 'information density' of a sub-region, assuming that all amino acids are informationally equivalent. Sander and Schneider (21) calculate a variation entropy for each column. Brouillet *et al*. (22) calculate the mean and standard deviation of the pairwise distances between amino acids in each column of an alignment, using a 20×20 distance matrix. None of these methods were found to be ideal for incorporation into Clustal X. Apart from the latter, none of the methods use a standard residue exchange matrix, as is needed for consistency with the alignment process, as well as providing a natural way to allow the user to customise the quality analysis by varying the matrix. The advantage of the geometric interpretion developed for Clustal X is that statistical methods can then be applied to define a mean value for the column and distances can be measured between each sequence and the mean: upper limits for the expected distance between any residue and the mean value can be defined and thus exceptional residues can be identified.

Low-scoring segments in the sequences can also be highlighted in the alignment (Figs 1 and 2). Low-scoring segments most often result from one of three major causes: high divergence between the sequences; errors in input sequences, most notably frameshifts; and misalignments. If the cause can be ascribed to high divergence, the alignment may not be wrong, but should be regarded as unreliable in the low-scoring segment. In particularly unreliable segments, CLUSTAL X may mark out every sequence! The alignment in such a region is likely to be meaningless. Frameshift errors are more frequent than usually realised. In the alignment of EFTUs taken from Swiss-Prot release 34, four sequences have short frameshifts within the region shown in Figure 1. Suspect sequences can be investigated with frameshifting alignment programs such as PairWise in WiseTools (24), or Framesearch in the GCG package. It is important to detect and remove sequences containing errors, as they confound many types of inferences based on multiple alignments, and may themselves also cause the propagation of further alignment errors.

We have found the low scoring segments test to be remarkably powerful, picking up a number of frameshifts and leading to the correction of many misalignments. Not every highlighted region is false but, by checking them over, the major errors are almost always uncovered. Nevertheless, there are situations where the test may give a false sense of alignment accuracy. This could happen when aligning sequences with strong amino acid residue biases ('reduced sequence complexity'). Tandem repeats are another case, since superposition of the wrong repeats could still give a high scoring alignment. Alignments of highly divergent membrane proteins are tricky on both counts since there are many transmembrane helices with hydrophobic amino acid biases.

More specialised, detailed alignment analysis programs are available (24–27). The advantages of CLUSTAL X are that the quality analyses are very fast as well as being integrated into the alignment package and the results are displayed graphically on the screen, with any low-scoring regions highlighted by shading the alignment background. This interactive system provides an efficient and flexible approach to alignment analysis and correction.

## Correcting misaligned regions

In Figure 2, a 'model' protein misalignment has been set up. For clarity, the closely related EFTU sequences have been deliberately misaligned. Genuine misalignments would normally be highly divergent with only a few identities in particularly conserved columns. In such cases, if the correct alignment can be ascertained, this may be by matches between residue similarities rather than identities.

In the example, a misaligned segment of EFTU_ECOLI is first detected and marked by applying the low-scoring segments algorithm (Fig. 2A). Next, a region of the alignment spanning the error is selected using the cursor. The menu option 'reset all gaps before alignment' is toggled on: in this example there are falsely inserted gaps that must be deleted. This is not always the case, and if the existing gaps seem correct, the option can stay switched off. Now the 'realign selected residue range' option is invoked. The misaligned region is now rapidly and correctly aligned again, and the false gaps are deleted (Fig. 2B). This time the low-scoring segments algorithm finds only short segments ascribable to natural sequence divergence. Realignments in which the gaps are left in may result in columns with nothing but padding characters, in which case there is a menu option available to delete these.

The realignment process uses the alignment parameter default settings, or as they are set up by the user. Misaligned regions are often more divergent than other regions of the alignment, which means that the alignment score may not be much higher than misaligned alternatives. Therefore it may be necessary to lower gap penalties to allow the sequences to align: this is tested by trial and error. However, the user should be aware of two factors that already affect the gap penalties in the local realignment. There is no gap penalty at the ends of a selected region, so it is free to put new gaps there: judicious selection of the range boundaries can direct gaps to desired sites. Gap penalties are also lowered at existing gaps if these are retained. These factors mean that the selected range may give a better alignment without having to lower the gap penalties.

## Further uses for the low scoring segments

In CLUSTAL X, the new algorithm for marking low-scoring segments has been implemented for visual interaction. However, the algorithm has the potential for wider usage. There are currently many projects to automatically produce databases of

multiple sequence alignments. The alignments tend not to be of high quality as it has been difficult to distinguish good and bad aligned regions rapidly and reliably. Removing sequences with low-scoring segments below a cut-off score should dramatically improve these alignments, as all sequences that contain major errors, or are too divergent to align, can be trapped.

The algorithm also has the potential to automatically establish the domain boundaries in sets of partially-related multi-domain proteins. In this case the Smith–Waterman best local alignment algorithm, finding the approximate regions encompassing the homologous domains, would be harnessed to the forwards-backwards approach, summing both the positive and negative scoring segments in order to define sharp boundaries. A simpler application would be end-trimming in an alignment, since the termini of proteins are often poorly conserved.

## ACKNOWLEDGEMENTS

J.T.was supported by institute funds from INSERM, CNRS and the Ministère de la Recherche et Technologie and the EMBL. We thank the many users of CLUSTAL W who have reported bugs/suggestions, and those who beta tested CLUSTAL X. We would also like to thank Dino Moras, Kevin Leonard, Matti Saraste and Frank Gannon for support during this work.

## REFERENCES

1 Feng,D.F. and Doolittle,R.F. (1987) *J. Mol. Evol.*, **25**, 351–360.
2 Taylor,W.R. (1988) *J. Mol. Evol.*, **28**, 161–169.
3 Stockwell,P.A. and Peterson,G.B. (1987) *Comput. Applic. Biosci.*, **3**, 37–43.
4 Thirup,S. and Larsen,N.E. (1990) *Proteins*, **7**, 291–295.
5 Clark,S.P. (1992) *Comput. Applic. Biosci.*, **8**, 535–538.
6 De Rijk,P. and De Wachter,R. (1993) *Comput. Applic. Biosci.*, **9**, 735–740.
7 Galtier,N., Gouy,M. and Gautier,C. (1996) *Comput. Applic. Biosci.*, **12**, 543–548.
8 Higgins,D.G. and Sharp,P.M. (1988) *Gene*, **1**, 237–244.
9 Higgins,D.G., Bleasby,A.J. and Fuchs,R. (1992) *Comput. Applic. Biosci.*, **8**, 189–191.
10 Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) *Nucleic Acids Res.*, **22**, 4673–4680.
11 Higgins,D.G., Thompson, J.D. and Gibson,T.J. (1996) *Methods Enzymol.*, **266**, 383–402.
12 Vingron, M. and Sibbald, P.R. (1993) *Proc. Natl. Acad. Sci. USA*, **90**, 8777–8781.
13 Henikoff,S. and Henikoff,J.G. (1994) *J. Mol. Biol.*, **243**, 574–578.
14 Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) *Comput. Applic. Biosci.*, **10**, 19–29.
15 Benner,S.A., Cohen,M.A. and Gonnet,G.H. (1994) *Protein Engng*, **7**, 1323–1332.
16 Notredame,C. and Higgins,D.G. (1996) *Nucleic Acids Res.*, **24**, 1515–1524.
17 Gotoh,O. (1996) *J. Mol. Biol.*, **264**, 823–838.
18 Parry-Smith,D.J. and Attwood,T.K. (1991) *Comput. Applic. Biosci.*, **7**, 233–235.
19 Zvelebil,M.J.J.M., Barton,G.J., Taylor,W.R. and Sternberg,M.J.E. (1987) *J. Mol. Biol.*, **195**, 957–961.
20 Smith,R.F. and Smith,T.F. (1990) *Proc. Natl. Acad. Sci. USA*, **87**, 118–122.
21 Sander,C. and Schneider,R. (1991) *Proteins Struct. Funct. Genet.*, **9**, 56–68.
22 Brouillet,S., Risler,J.L., Henaut,A. and Slonimski,P.P. (1992) *Biochimie*, **74**, 571–580.
23 Birney,E., Thompson,J.D. and Gibson,T.J. (1996) *Nucleic Acids Res.*, **24**, 2730–2739.
24 Schuler,G.D., Altschul,S.F. and Lipman,D.J. (1991) *Proteins Struct. Funct. Genet.*, **9**, 180–190.
25 Vingron,M. and Argos,P. (1991) *J. Mol. Biol.*, **218**, 33–43.
26 Friemann,A. and Schmitz,S. (1992) *Comput. Applic. Biosci.*, **8**, 261–265.
27 Livingstone,C.D. and Barton,G.J. (1993) *Comput. Applic. Biosci.*, **9**, 745–756.