JavaShade: multiple sequence alignment
box-and-shading on the World Wide Web

Mark R. Southern\(^1\) and Alan P. Lewis\(^2\)

\(^1\)Bioinformatics, Research Information Systems, and \(^2\)Advanced Technology and Informatics Unit, GlaxoWellcome Medicines Research Centre, Stevenage, Herts SG1 2NY, UK

Abstract

Results: JavaShade is a multiple sequence alignment box-and-shade tool for generating high quality printed output that uses a variety of methods for boxing and shading, allowing the most appropriate functions to be chosen for displaying the most meaningful positions in an alignment.

Availability: JavaShade is available from the WWW at http://industry.ebi.ac.uk/JavaShade

Contact: ap111418@ggr.co.uk

Introduction

JavaShade is a multiple sequence alignment box-and-shade tool that has been designed to generate publication-quality Postscript output in a user-friendly, interactive, flexible manner. It is written as a Java 1.0.2 applet so as to be compatible with the majority of Web browsers. It has been tested, and works consistently in Java enabled version 3 and 4 browsers on win95, winNT and SGI platforms.

Visualisation can provide one of the more powerful means of analysing sequence conservation across a multiple sequence alignment, condensing the mass of information present, and allowing the rapid identification of regions of possible structural and/or functional importance (Livingstone and Barton, 1993; Zvelebil et al., 1987). Such observations can quickly locate target residues for site-specific mutagenesis experiments, especially when combined with other data, such as secondary structure and residue exposure predictions.

One of the most informative ways of representing a multiple sequence alignment is by boxing and shading the residues therein. Residues can be highlighted manually to draw attention to specific features of the alignment. Alternatively, they can be highlighted in a defined manner, for example, to display the degree of sequence conservation across the alignment. Boxed and shaded alignments are commonly found in publications.

Currently there are several software tools that can box and shade multiple sequence alignments, offering a variety of functions. Examples are AMAS and Alscript (Livingstone and Barton, 1993; Barton, 1993), Lasergene (DNASTAR), PrettyPlot, PrettyBox (http://cmgm.stanford.edu/EGCG-doc), Shadybox (http://www.angis.su.oz.au/%7echuynh/ShadyBox.html), and Pfam Alignment Viewer (http://genome.wustl.edu/Pfam/java/ReleaseNotes_avlbl.html).

However, in one way or another they do not fulfil our requirements of being user-friendly, cross-platform, interactive, freely accessible, generating publication quality output, and having a variety of biological approaches to sequence conservation. For example, Shadybox is only available on Unix platforms, and performs simply percentage identity discrimination; PrettyPlot is command line driven and output cannot be viewed interactively; Pfam Alignment Viewer cannot generate output for printing, does not have user-defined boxing and shading, performs simply percentage identity discrimination, and the available MSF files for input must reside on the server.

Algorithms and implementation

JavaShade reads GCG MSF file format alignments (Devereux et al., 1984) a common format that most multiple sequence alignment programs can output. Java 1.0.2 applets are prevented from reading and writing to the file system of the local machine on which the applet is running and thus the current method of entering an MSF alignment is to cut-and-paste the alignment into the text box of the applet.

The residues in the displayed alignment (Figure 1) can be altered to a variety of user-defined styles using both simple mouse drags, or by using a number of different available functions. The functions act upon user-selected sequences, rather than the whole alignment, to allow shading of different sequences using different functions (Figure 1). In addition to simple percentage identity, and selection of individual residue types, pair-wise and column based conservation algorithms have been implemented.

The ‘Zvelebil’ function (proteins only) implements a column-based comparison (Zvelebil et al., 1987) of the residues present at each position within the sequence alignment. Each amino acid is assigned a yes or no value to each of ten chemical properties, where gaps and unknown residues are
Fig. 1. Postscript output from JavaShade. A protein multiple sequence alignment of a number of CD8 orthologues has been shaded at increasing intensities using the ‘Matrix’ function to show the sequence conservation across the alignment. The secondary structural elements are also highlighted by shading on the lcd8_struct line.

modelled as having a yes value for each of the properties. If a position in the alignments is invariant then the conservation number \( C_i \) equals 1.0. If a position is variant, each property of the amino acids present at that position is considered. A count, \( P \) is incremented for each chemical property where a difference is observed. The conservation number \( C_i \) is then defined as:

\[
C_i = 0.9 - (0.1 \times P).
\]

If \( P = 10 \) then \( C_i \) is set to 0 rather than \(-0.1\). If the value of \( C_i \) for a particular column is greater than or equal to a user defined cut-off, then that column is altered to the current box and shade style.

The ‘Compare’ function (proteins, but can also be applied to nucleic acid when the cut-off is equivalent to identity) performs pair-wise comparisons for each residue of a single user-defined sequence against the corresponding residues of the remaining selected sequences based on a Dayhoff matrix (Schwartz and Dayhoff, 1979) as described by Gribskov et al. (1987). Matched (or unmatched, depending on selection) residues will be altered to the currently defined box and shade style.

The ‘Matrix’ function (proteins only) implements pair-wise comparisons of residues at each position within the alignment. At each position within an alignment, the scores from the Dayhoff matrix for each residue are added together, and the sum is divided by the number of sequences. The final scores are thus an averaged representation of all the residues present. This ‘virtual’ amino acid is compared to each of the residues within the column, and if the score is greater than or equal to a user-defined score cut-off, a count, \( P \) is incremented. The count \( P \), for a position within the alignment is divided by the number of sequences. If this value is greater than or equal to a user defined percentage cut-off then those residues that meet the score cut-off are altered to the current box and shade style.

Output from JavaShade is in the form of customisable publication quality postscript. Due to the file system access restrictions in Java 1.0.2, the postscript document is generated dynamically on the client, saved back to the server on which the applet resides, and then downloaded to the client via http. Postscript format has the benefit of including multiple page facilities and, in addition, there are no problems with resolution on different platforms.

Since the JavaShade applet is loaded dynamically across the WWW, the latest version will always be available from the URL given in the abstract of this manuscript. Possible improvements to JavaShade include improved graphics performance, increased support for printing options (more fonts, customisation), an undo feature, support for additional multiple alignment file formats, the ability to access a multiple alignment file from across a network in addition to the cut-and-paste entry method, inclusion of further protein matrices for calculation of variability using the ‘Matrix’ function, and additional algorithms for determination of regions for box-and-shading (e.g. charge, hydrophobicity).

Acknowledgements
We would like to thank BBSRC for funding part of this work.

References
We would like to thank BBSRC for funding part of this work.


