

# Alignment Visualization

August 14, 2002

Sequence Analysis & Consulting Service

Seminar Series

Susan Johns

(johns@cgl.ucsf.edu)

How do I visualize a multiple sequence alignment and then create a publishable hard copy output of it?

**Multiple sequence alignments** have many uses in sequence analysis [protein modelling, structure prediction, molecular evolution studies, and sequence pattern or motif detection].

In all of these functions, the underlying purpose is to display or summarize the relationships between a group of sequences.

The significance of an alignment depends on why one is generating it.

**Multiple sequence alignments** can be produced by a number of different computer program, by editing. the file manually

The same program can produce different results, depending on the scoring matrix used for the alignment.

There is no **standard format** for a multiple sequence alignment. That depends on the program used to generate the alignment.

Conversion software can sometimes be used to go from one multiple sequence alignment format to another.

**Multiple sequence alignments** can be visualized with a number of different programs. Each has its own combination of parameters that can be modified to change the nature of the output.

Sometimes additional post processing is required to get the desired results.

**Never** rule out the need to hand-edit an output file to get exactly what you want.

A series of **multiple sequence alignments** were made using the same data set.

Depending on the scoring matrix and program used, the resulting alignments varied in length from 147 to 368.

[note: not all possible combinations of scoring matrices and programs were used to come up with this range]

A user must look at all these alignments and decide which made the most sense and was true to the data known about the sequences in question.

In this case, all have a cytochrome c binding site (cxxch) followed by a M that binds to the heme group.

A number of different programs that can be used to generate multiple sequence alignments. Some are found locally and others are on the web.

- blocks
- clustalw
- multalin
- pileup
- pima

Whatever the source of the alignment, the user needs to assess what changes, if any, need to be made in the alignment file in order for it to be used in the desired visualization tool.

## blocks

The Match-Box software proposes protein sequence alignment tools based on strict statistical criteria. The method circumvents the gap penalty requirement: in the Match-Box method, gaps are the result of the alignment and not a governing parameter of the matching procedure. A reliability score is provided below each aligned position. The Match-Box program is particularly suitable for finding and aligning conserved structural motifs, in particular in protein core.

## sites

### BCM Search Launcher

[<http://dot.imgen.bcm.tmc.edu:9331/multi-align/multi-align.html>]

[select the **Match-Box option** from the email listing to get a full alignment back]

### Match-Box Web Server

[full option web site]

[[http://www.fundp.ac.be/sciences/biologie/bms/matchbox\\_submit.html](http://www.fundp.ac.be/sciences/biologie/bms/matchbox_submit.html)]

input formats: FASTA, HSSP, MSF

matrices available:

BLOSUM series [BCM - BLOSUM62 ]

PAM series

Gonnet series

Johnson series

output formats:

plain text [BCM]

postscript file

FASTA format file

MSF format file

HSSP format file

## clustalw

CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice.

### sites

#### on socrates

- Unix command line version
- New SACS web page

#### other sites

##### BCM Search Launcher

[<http://dot.imgen.bcm.tmc.edu:9331/multi-align/multi-align.html>]

##### European Bioinformatics Institute

[<http://www2.ebi.ac.uk/clustalw/>]

input formats: same as output

### matrices available:

- BLOSUM series
- PAM series
- Gonnet series
- Identity matrix
- User defined

### output formats:

- clustalw format (.aln file)
- NBRF/PIR format (.pir file)
- GCG/MSF format (.msf file)
- PHYLIP format (.phy file)
- GDE format (.gde file)

## multalin

Multalin creates a multiple sequence alignment from a group of related sequences using progressive pairwise alignments with hierarchical clustering.

### sites

#### on socrates

- Unix command line version
- SACS web interface

#### other sites

##### MultAlin web site

[<http://www.toulouse.inra.fr/multalin.html>]

input formats: MultAlin/FASTA, GenBank\*, SwissProt\*

### matrices available:

- BLOSUM62
- Dayhoff
- Risler
- Genetiq
- DNA, AltDNA
- Identity

### output formats:

- gif file (France)
- mul/FASTA format (.mul file)
- MSF format (.msf file)
- doc format (.doc file)

## pileup

PileUp creates a multiple sequence alignment using a simplification of the progressive alignment method of Feng and Doolittle .

## sites

### on socrates

- Unix command line version
- SACS web interface
- W2H interface

input formats: GCG formatted files

### matrices available:

- BLOSUM series
- PAM series
- structural

### output formats:

- GCG/MSF format (.msf file)

## pima

pima performs a multi-sequence alignment of a set of (presumably related) sequences using an extension of our covering pattern construction algorithm (Smith and Smith 1990, 1992). All pairwise comparisons between sequences in the set are performed and the resulting scores clustered into one or more families using two different linkage rules: 1) maximal linkage and 2) sequential branching.

### sites

#### on socrates

- Unix command line version

#### other sites

##### BCM Search Launcher

[<http://dot.imgen.bcm.tmc.edu:9331/multi-align/multi-align.html>]

input formats: FASTA

### matrices available:

- pattern based - does not apply

### output formats:

- FASTA format (.pima file)
- [two files are created one for each type of linkage rule]

A number of different programs that can be used to generate visual outputs from multiple sequence alignments. Again, some are found locally and others are on the web. Only local ones will be covered here.

## alscript

### boxshade

### mview

### prettybox

Whatever the visualization mode, the user needs to assess what is to be emphasized in the alignment file in order to use the desired visualization tool properly.

**Alscript** takes a multiple sequence alignment in AMPS block-file format and a set of formatting commands and produces a postscript output file. [Complete coloring control with a large learning curve.]

### input formats:

- AMPS file
- clustal file
- msf file

controllable characteristics:

- bold text
- font size
- orientation [landscape or portrait]
- shading colors [as many as you wish to define]

output:

- postscript file

**Alscript** is a very powerful program that allows users to do many things. Unfortunately, it also has a high learning curve to become proficient with the software.

If your alignment file has any tildes in it, these must be removed by using **tr** on your msf file.

```
tr ~ \. < hits2.msf > hits2b.msf
```

Using **alscript** is a three-step process. First, a blocks file needs to be generated. For an msf file, the program that does this is **msf2b1c** and for a clustal file, the program is **clus2b1c**.

Then a control file needs to be created that contains the information for formatting the desired output file. A user consults the **alscript** user's guide to get a handle on the types of commands to use.

With the control file created, the program is run to generate an output file. This is usually an iterative process, with the user changing the control file to fine-tune the output.

**Boxshade** is a program intended for shading multiple aligned sequence files.

input formats:

- clustal file
- msf file
- phylip

controllable characteristics:

- block size
- displaying consensus
- font size
- orientation [landscape or portrait]
- plurality -[minimum number to match to be colored]
- shading colors [two settings - identical and similar]
- width

output:

- html file
- pict file
- postscript file

**Boxshade** is a program that prompts the user for the parameters that can be changed. There is a parameter file that sets up the default conditions for each type of output device. This file can also be customized if necessary.

**Mview** is a tool for converting the results of a sequence database search into the form of a colored multiple alignment of hits stacked against the query. Alternatively, an existing multiple alignment can be processed.

input format:

- clustalw
- fasta
- hssp
- msf
- pir
- plain

controllable characteristics:

- bold text
- displaying consensus
- number placement [labels on left can be controlled]
- plurality -[minimum number to match to be colored]
- shading colors [default is by residue type]
- width

output:

- html file

**Mview** is a program that has a complex command line syntax. The user needs to use a help file and example command lines to get it to work.

```
mvview -in msf -html head -css on -bold -coloring  
consensus -threshold 50 -con_gaps on -ruler off  
-consensus off -con_coloring identity -width 60  
hits2.msf > hits2-mvview.html
```

There is no indication that the program is working other than the return of the prompt.

**Prettybox** is the GCG multiple sequence alignment visualization tool.

input format:

- msf

controllable characteristics:

- block size
- bold text
- displaying consensus
- font size
- matrix used for similarity scoring
- number placement [right or left side]
- orientation [landscape or portrait]
- plurality -[minimum number to match to be colored]
- shading colors [four settings - only three used]
- width

output:

- postscript file

**Prettybox** is a program with both command options and interactive user prompts. When running the program on socrates, the default output is a postscript file. The SACS web interface to the program prompts the user for the name of the closest postscript printer and automatically prints the output file there.

When the default setting output takes up just too many pages, look at the prettybox single page hints to solve the problem.

To show the characteristics of each viewer program the following set of conditions were set up.

desired settings

same alignment file used (hits2.msf)  
approx. 50% of a column needed to be the same residue in order to be colored (in our case 9 of the 17).  
no consensus

Comparison of results:

### alscript

advantages:

as many colors as you wish to define  
complete control over text highlighting  
ability to draw boxes around text

disadvantages

no control over the number of characters on a line  
large learning curve to master program  
doesn't accept tildes, must change input alignment

### boxshade

advantages:

color options for identity and similar residues

disadvantages:

might not agree with what is considered to be similar  
small font size

### mview

advantages:

colored output

disadvantages:

no real control over coloring scheme  
complex command line syntax  
only produces html output

### prettybox

advantages:

similarity matrix selection  
three shading options for highlighting

disadvantages:

no color selection

Matching the generated alignment file to a visualization tool requires some information on what tools will accept what alignment format. All four tools will accept the **msf** format.

### blocks

(plain text - back from web site)

### clustalw

.aln file can be used directly in alscript, boxshade and mview.

.aln file into msf conversion

.msf file - non-standard  
conversion to standard format

### multalin

.msf file is non-standard  
conversion to standard format

### pileup

output works fine as is

### pima

fasta formatted alignments files  
conversion to msf format

The **BCM blocks** output on the web contains all the information needed for further analysis, however, this data requires a lot of work to get it into shape.

- 1) manually create a fasta formatted alignment file
  - a) Pare down the raw data to just the alignment section. Remove all the numbering at the top of each section.
  - b) Edit this into a fasta formatted file by putting the segments of each sequence back together and then removing the numbers from any line after the first one.
  - c) Put a ">" symbol in front of the number on the first line. Move to the space to the right of the number and press the return key. A section of the file should now look like this.

```
>1
-----QDGDAAkgekefnkckachmiqapdgtDIKGGKTGP-----
--nlygvvgrkiaseegfkygegiLEVAEKNPDLTWTREADLIEYVTDPKPWLVRMTDDRGAKTRMTFKMG
KNQADVVAFLAQNSPDAGGDGEAA
```

- 2) Replace the "-" symbol with a "." by using the **<b>tr</b>** command.

```
tr \- \. < try.align > try.align2 <rtm>
```

- 3) Run the file through the readseq program to convert it from a fasta formatted alignment file to an msf alignment file.

- 4) Use the editor to fix the **type** of the alignment.

```
cytc-b.msf MSF: 164 Type: N January 01, 1776 12:00 Check: 182 ..
```

Change to

```
cytc-b.msf MSF: 164 Type: P January 01, 1776 12:00 Check: 182 ..
```

The file is now ready to use.

The **clustalw** .aln format can not be read by readseq. It would be easier to just re-run the program and this time select msf as the desired output format. However, if that is not possible, here are the steps to convert an .aln file into a msf one.

- 1) manually create a fasta formatted alignment file
  - a) Pare down the data file to just the alignment lines. Remove the title from the top and the consensus line from the bottom of each section.
  - b) Edit this into a fasta formatted file by putting the segments of each sequence back together and then removing the identifier name from any line after the first one.
  - c) Put a ">" symbol in front of the identifier name on the first line. Move to the space to the right of the number and press the return key. A section of the file should now look like this.

```
>CCAUG
-----IDINNGENIIFTAN-CSACHAGG-----NN-----VIMPEKTLKKDALADAN
-----KMVSVNAITY-----QVTNGKN-----AMP-----AFGS-----
-RLAETDIEDVANFVLTQSDKGWD-----
```

- 2) Replace the "-" symbol with a "." by using the **tr** command.

```
tr \- \. < try.align <try.align2 <rtm>
```

3) Run the file through the readseq program to convert it from a fasta formatted alignment file to an msf alignment file.

4) Use the editor to fix the **type** of the alignment.

```
cytc-cwb2.msf MSF: 164 Type: N January 01, 1776 12:00 Check: 182 ..
```

Change to

```
cytc-cwb2.msf MSF: 164 Type: P January 01, 1776 12:00 Check: 182 ..
```

The file is now ready to use.

The **clustalw** msf format is a non-standard version of the format.

1) Run the file through the readseq program to convert it to a better msf alignment file.

2) Use the editor to fix the **type** of the alignment.

```
cytc-b.msf MSF: 164 Type: N January 01, 1776 12:00 Check: 182 ..
```

Change to

```
cytc-b.msf MSF: 164 Type: P January 01, 1776 12:00 Check: 182 ..
```

The file is now ready to use.

The **multalin** msf format is a non-standard version of the format, however, it can easily be corrected by simple editing.

1) Remove the lines at the top of the file until the **MSF:** line is reached. Also remove the **Consensus** lines from the file.

2) On the **MSF:** line insert the term **Len:** with a space separating it and the number on that line. Insert on that line after the number the term **Type: P**. Have a space between this term and the next one on the line.

The file is now ready to use.

The **pima** standard output is a fasta formatted alignment file. In this case the program produces two of them. Select the one that meets your needs.

1) Remove the title of the section and the pattern sections from the top of the data file. Pattern sections start with the term **>pat-**.

2) Run the modified file through the readseq program to generate an msf formatted alignment file.

3) Replace the "-" symbol with a "." by using the **tr** command

```
tr \- \. < try.align >try.align2 <rtm>
```

4) Use the editor to fix the **type** of the alignment.

```
cytc-p.msf MSF: 164 Type: N January 01, 1776 12:00 Check: 182 ..
```

Change to

```
cytc-p.msf MSF: 164 Type: P January 01, 1776 12:00 Check: 182 ..<
```

The file is now ready to use.

Even if you have an output file, it might not be in the proper shape for your final purpose. If you want to have all of the alignment on one page or as few pages as possible, it may require some playing with the viewer program's options.

[boxshade hints page](#)  
[prettybox single page hints](#)

The name of the sequences are taken from their input files and rarely are what you would like them to be. Be prepared to go into the generated postscript files and change these names to something more acceptable.

Log into your socrates account by clicking on the socrates icon in the window on the left-hand side of the screen and responding with your account name and password.

Now, copy over to your account the necessary data. Enter the following commands. The first one does the actual copying. And the second one moves you into this area.

```
viewer_setup<rtm>  
cd viewer_analysis<rtm>
```

As a result of these commands, you are now in a sub-directory called **viewer\_analysis** where the necessary files are located. Activate the gcg software.

For a hands on session with this type of information, a number of files have been copied over into your account.

The first part of this process is usually the converting an alignment file into a viewer usable form. Two files, **pima-ex.alignment** and **clustalw-ex.alignment** were copied for this purpose. Use the links below to have the instructions on the browser screen while you work in the socrates window. Do both conversions.

Part two of this process is actually running a viewer to obtain some visualized alignment output. Two of the viewers are really easy to use. These are **boxshade** and **prettybox**. Use the file **try-unfixed.msf** and the links to the screen traces to each program given below to create some output.

Print a copy of your results by using the following command. The name of the printer in the seminar room is **class**.

```
lpr -Pclass your_output_file_name<rtm>
```

Normally running the **alscript** program is a three- or four-step process. First, the msf file is checked to see if it contains any tildes. If so, it is changed. Next, the program **msf2blc** is run to generate the needed blocks file. The user then starts to create a control file containing the input to be used in the program. Finally, the program is run using the control file to see if it works and what the output looks like. For this class this has all been taken care of. You will be modifying a control file that works to change the colors around a known motif in the aligned sequences.

Make the following changes with the **pico** editor in the **try.als** file to change the color of the most frequent residues in columns 38, 41 and 42 to magenta rather than their current red. Then, run the program to make sure your changes work.

**mview** has a very complex command line syntax. Copy the command line fragments given below one at a time onto the socrates prompt. Be sure to have a space between each fragment. When they are all copied, press the **return key** and run the program.

```
mview -in msf -html head -css on -bold -coloring consensus
-threshold 50 -con_gaps on -ruler off -consensus off
-con_coloring identity -width 60 try.msf > try-mview.html
```

Check the contents of your working directory with the **ls -la** command to see that the desired output file has been created.

Change the name of this file to your\_account\_name-mview.html and then copy it over to the following spot in the SACS web site.

```
cp try-mview.html johns-mview.html <rtn>
cp johns-mview.html /usr/local/html/sacs/home/johns/class <rtn>
```

Use the mouse to click on the Netscape window, then select **Open** from the **File** option menu. Select **Location in Navigator** and enter the following name in the window that appears.

<http://www.sacs.ucsf.edu/home/johns/class/johns-mview.html>

When you are finished looking at your html file, click on the **Back** button at the top of the Netscape window to return to this page.

Generating an output file doesn't ensure that what you get is what you want to have. There are times when you want to have your output all on one page, be a certain font size so that it can be seen from a given distance, or perhaps be colored. This section of the seminar deals with these questions.

### Fitting on one page

The general idea here is to figure out what size font and page orientation allows your alignment to fit on the page.

Use the **mshr.msf** file to create a one page output with either the boxshade or prettybox program. Print a copy of your results by using the following command. The name of the printer in the seminar room is class.

```
lpr -Pclass your_output_file_name<rtn>
```

### Large print

Sometimes a larger than usual font is needed. This situation is just the reverse of the previous one. Use the standard settings and experiment until you get the conditions you want.

In the example given below, the following command line switches were used.

**-fat** makes the font bold  
**-font=20** changes the font size to 20  
**-wid=30** changes the number of alignment characters shown in a line (default is 60)  
**-num=N** removes the numbering at the end of the line  
**-nohea** removes the normal heading from each page of output

The result of these changes was displayed at the beginning of the seminar session.

### Coloring output

Color can make your output more appealing to the eye. A perl script is in development that allows the coloring of prettybox generated postscript files. The name of this script is **color-pb**.

Select one of your prettybox generated postscript files and run it through the script. The colors cyan, orange and yellow produce nice shades of gray when printed on a gray scale printer. In normal operation the color value for dark is not used, but all the coding is there in case you want to edit the file again and put in additional colored areas.

### Alscript

Barton, G. J. (1993), "ALSCRIPT a tool to format multiple sequence alignments," *Protein Engineering*, 6, 37-40.

### Mview

Brown, N.P., Leroy C., Sander C. (1998). MView: A Web compatible database search or multiple alignment viewer. *Bioinformatics*. 14(4):380-381.

### Prettybox (part of the GCG package)

PrettyBox was written by Rick Westerman of Purdue University. Copyright (1998) by Purdue Research Foundation, West Lafayette, IN 47907. All Rights Reserved.

### boxshade server:

• [http://www.ch.embnet.org/software/BOX\\_form.html](http://www.ch.embnet.org/software/BOX_form.html)

Boxshade has been written by Kay Hofmann and Michael D. Baron, there is no publication to cite.

