pyBoxshade

A desktop version of BOXSHADE(https://embnet.vital-it.ch/software/BOX_form.html)

Purpose

pyBoxshade is a program for creating good-looking printouts from alignments of multiple protein or DNA sequences. The program does no alignment by itself, it takes as input a file pre-processed by a multiple alignment program or a multiple sequence editor.
In the program output, identical and similar residues in the multiple alignment are represented by different colours of letters or shadings (colours of background). There are many options concerning the kind of shading to be applied, whether to include a ruler line, sequence numbering, a consensus line and so on. One of the main changes made during construction of this version was to provide access to these options through dialogs. The original program (BOXSHADE) simply reads in a sequence, processes it, then quits. **pyBoxshade** holds the sequence in memory until a new sequence is read in, or the user quits. Any number of different outputs can be done with the same sequence; the output can be viewed dynamically before a version is stored as a file.

Input formats

pyBoxshade supports a number of different input formats; it uses the BioPython library for reading alignment files, and can therefore read most of the formats supported by that library. Currently this includes Clustal format (.aln), FASTA format, Phylip format (interleaved or sequential), MSF, Nexus and Stockholm formats. The program attempts to determine the file type, so it should handle all of these transparently.

Output formats

pyBoxshade provides four types of output, those I thought would be of most use:
1. PS (PostScript) files for printing directly or further conversion. I have had good success opening these files and converting them to PDF, TIFF or other formats with Preview (on Mac), IrfanView (Windows) or GIMP (either platform).
2. RTF (Rich Text Format) for export to various word-processing and graphics programs (seems to work in TextEdit (Mac OS), Microsoft Word or OpenOffice).
3. PNG (Portable Network Graphics). This format can first be viewed on screen, then saved as an image file. It is a pixel-based image format (similar to TIFF or JPEG), so is not suitable for enlarging or where high resolution images are required. In the latter case it is possible to make a larger image using a large font and shrink this image down to the required size.
4. ASCII output showing either the conserved residues or the varying ones.

For formats 1-3 the font size can be selected, and for PS output one can specify Portrait or Landscape page orientation. **NB** Because of limitations of PostScript, the image will be clipped to the size of an A4 page, if it is too wide. The program does a check to see if the image that has been asked for will fit, and offers the user the option of proceeding anyway or cancelling the operation.

Shading strategy (similarity to consensus or single sequence)

The shading algorithm used by BOXSHADE (and hence by **pyBoxshade**) is completely configurable by the user, and is not based on any specific mutational table. Firstly, in order for there to be a consensus of any kind at a position, a threshold fraction of the sequences must agree. This threshold fraction can be any number between 0 and 1. The number of sequences that must agree for there to be a consensus is, as you might expect, this

fraction times the total number of sequences in the alignment, rounded to the nearest whole number.
An additional option for this kind of consensus is to apply a different colouring/shading where all sequences have the same residue (globally conserved).
If an identity-type consensus is found, and similarity shading is in operation, the program looks to see if the remaining residues at that position are similar to the consensus residue. The amino acids that are to be considered similar to the consensus residue are defined in the '**Sims**' dialog. In this dialog,

S |TA |
means that both **T** and **A** are considered similar to **S**, where there is a conserved S residue in more than threshold number of sequences. However, it does NOT mean that T and A are similar to each other. This would have to be specified in the A or T box.

If there is no identity-type consensus, the program looks for a 'consensus by similarity'; this tries to take account of the situations where most of the sequences may have (for example) R or K at a position, but neither at a high enough level to pass the threshold fraction. If there is not a single residue that is conserved (greater than the threshold) at a position, the program looks for a 'group' of amino acids that fulfils the requirements. 'Groups' are defined in the '**Grps**' dialog. Users can tailor these to their personal prejudices, or base them on their favourite mutational frequency table. Any amino acid not listed is assumed not to be in a group. All members of a group are considered to be mutually similar, unlike the **Sims**, described above. If consensus by similarity is found, all the residues in the consensus group are shaded using the 'similar' shading defined by the user. If the user does not select 'shading by similarity', only the identity-type consensus is displayed.

Note that cases where two residues, or groups of residues, fulfil the threshold requirements (as could happen with values of the threshold fraction less than or equal to 50%) are treated as having no consensus.
As an alternative to a calculated consensus based on all the sequences in the alignment, the user can choose a ‘master sequence'. In this case the user specifies one of the sequences of the alignment and that sequence is taken to be the 'consensus'. Only those residues become shaded that are identical or similar to the chosen sequence. Output obtained with this option tends to be less shaded and neglects similarities between the other (non-chosen) sequences.

Consensus display

pyBoxshade offers the possibility to create an additional line holding a consensus sequence. The way this consensus line is displayed is controlled by specifying a string of exactly three symbols, in the 'chars to print consensus' box in the layout preferences dialog. As these symbols are not immediately intuitive, a brief explanation is necessary:

+ the first symbol is used for positions where there is no similar/identical relationship.
+ the second symbol is used for positions where a residue or group is identical, similar or a mixture of identical and similar, in greater than the threshold number of sequences of the alignment.
+ the third symbol represents positions that are identical in all sequences of the alignment.

For example, a parameter string " .*" (blank/point/asterisk) means: label all positions in the alignment with totally identical residues by an asterisk (*), all positions with greater than the threshold conserved residues by a point (.) and do not mark the other positions.

Besides points, asterisks and other symbols, there are three letters that act as special characters when they appear in the string: ‘B’, 'L' and 'U'. A 'B' can be used to mean a blank, an 'L' means that a lowercase representation of the most abundant residue at that position is to be used instead of a fixed consensus symbol while a 'U' means an uppercase character representation of that residue. A possible application would be the string BLU where conserved residues are represented by lowercase characters and identical by uppercase characters.

Sequence numbering

There is the possibility to add numbering to the output files. The numbers are printed between the sequence names and the sequence itself on the left hand side, or at the right hand side, or both, or neither. Since most of the input files either use no numbering or number the first position in the alignment always with a "1" (and that does not necessarily reflect the numbers within the original sequence), the user can specify the starting number for each sequence (the default is that all sequences start at 1).
pyBoxshade starts with the value entered for the first position and continues numbering every valid symbol, skipping blanks, '-','.' and ‘~’. If the user sets a negative start number, numbering passes straight from -1 to 1: there is no zeroth position.
Sequence numbering starts from the first residue. For sequences that do not start at the beginning of the alignment, or finish well before the end (i.e. are padded extensively at the left or right end of the alignment), numbering starts on the first line one which that sequence has a residue, and stops on the line that has the last residue of that sequence. At least, that is what it is supposed to do! Let me know if it doesn’t work for your alignment.

Marker line

pyBoxshade has the ability to print a marker or ruler line over the sequence alignment. This looks like this:

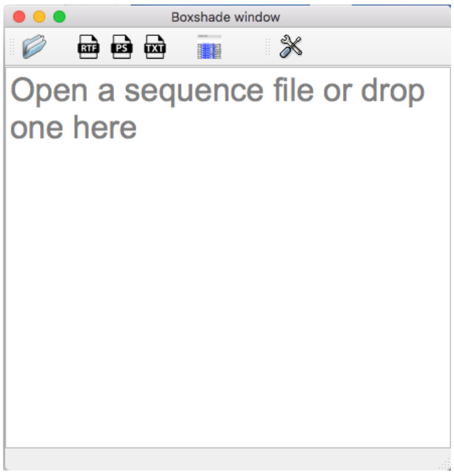
```
....:...10....:...20....:...30....:...40....:...50....:...60
AAAAACCCCCAAAAACCCCCGGGGGTTTTTGGGGGTTTTTCCCCCTTTTTGGGGGAAAAA
```

Numbers are right aligned with the residue in question, i.e. the right-most digit of the number is over the position indicated.

RUNNING THE PROGRAM

When **pyBoxshade** first starts, it will create a preferences file, in *user*/Library/Preferences on Mac OSX, or in the *user* area of the registry if on Windows. This should be transparent to the user.

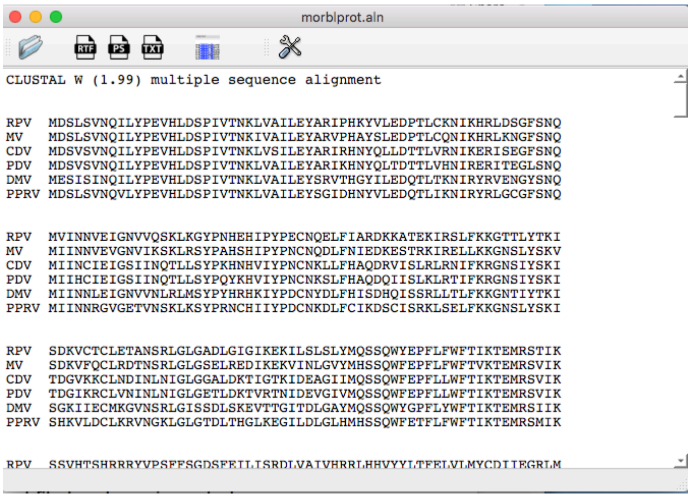
The main window looks like this:



Open a sequence file or drop one here

You can drop files onto the window and the program will attempt to open them. Note that it currently only deals with one alignment at a time, so it will open the first file of a group if you drop multiple files. If you want to open more than one alignment, you have to open them individually

(“Open”) opens a multiple sequence alignment file. This will be displayed in the main window, to allow the user to check the correct file has been imported, e.g.



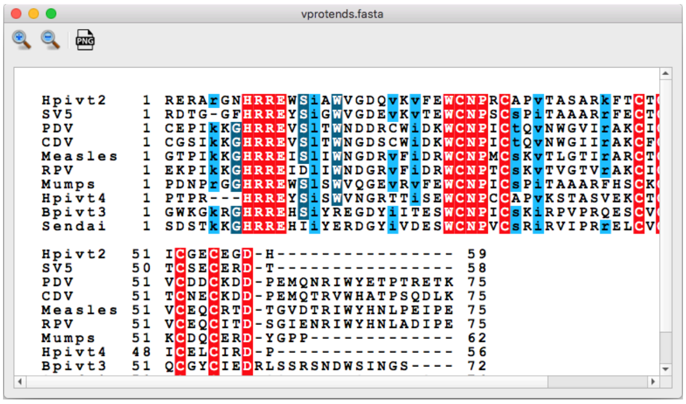
```
CLUSTAL W (1.99) multiple sequence alignment


RPV  MDSLSVNQILYPEVHLDSPIVTNKLVAILEYARIPHKYVLEDPTLCKNIKHRLDSGFSNQ
MV   MDSLSVNQILYPEVHLDSPIVTNKIVAILEYARVPHAYSLEDPTLCQNIKHRLKNGFSNQ
CDV  MDSVSVNQILYPEVHLDSPIVTNKLVSILEYARIRHNYQLLDTTLVRNIKERISEGFSNQ
PDV  MDSVSVNQILYPEVHLDSPIVTNKLVAILEYARIKHNYQLTDTTLVHNIRERITEGLSNQ
DMV  MESISINQILYPEVHLDSPIVTNKLVAILEYSRVTHGYILEDQTLTKNIRYRVENGYSNQ
PPRV MDSLSVNQVLYPEVHLDSPIVTNKLVAILEYSGIDHNYVLEDQTLIKNIRYRLGCGFSNQ


RPV  MVINNVEIGNVVQSKLKGYPNHEHIPYPECNQELFIARDKKATEKIRSLFKKGTTLYTKI
MV   MIINNVEVGNVIKSKLRSYPAHSHIPYPNCNQDLFNIEDKESTRKIRELLKKGNSLYSKV
CDV  MIINCIEIGSIINQTLLSYPKHNHVIYPNCNKLLFHAQDRVISLRLRNIFKRGNSIYSKI
PDV  MIIHCIEIGSIINQTLLSYPQYKHVIYPNCNKSLFHAQDQIISLKLRTIFKRGNSIYSKI
DMV  MIINNLEIGNVVNLRLMSYPYHRHKIYPDCNYDLFHISDHQISSRLLTLFKKGNTIYTKI
PPRV MIINNRGVGETVNSKLKSYPRNCHIIYPDCNKDLFCIKDSCISRKLSELFKKGNSLYSKI


RPV  SDKVCTCLETANSRLGLGADLGIGIKEKILSLSLYMQSSQWYEPFLFWFTIKTEMRSTIK
MV   SDKVFQCLRDTNSRLGLGSELREDIKEKVINLGVYMHSSQWFEPFLFWFTVKTEMRSVIK
CDV  TDGVKKCLNDINLNIGLGGALDKTIGTKIDEAGIIMQSSQWFEPFLLWFTIKTEMRSVIK
PDV  TDGIKRCLVNINLNIGLGETLDKTVRTNIDEVGIVMQSSQWFEPFLLWFTIKTEMRSVIK
DMV  SGKIIECMKGVNSRLGISSDLSKEVTTGITDLGAYMQSSQWYGPFLYWFTIKTEMRSIIK
PPRV SHKVLDCLKRVNGKLGLGTDLTHGLKEGILDLGLHMHSSQWFETFLFWFTIKTEMRSMIK


RPV  SSVHTSHRRRYVPSFFSGDSFEILISRDLVAIVHRRLHHVYYLTFELVLMYCDIIEGRLM
```

The other options from this window are:

RTF make RichText (RTF) file using the current parameters (the program will ask for the name of the file to save using the normal dialogs)

PS make Postscript (PS) file using the current parameters (the program will ask for the name of the file to save using the normal dialogs)

TXT create a text file using the current parameters (the program will ask for the name of the file to save using the normal dialogs) open a window to show an image drawn using the current parameters;

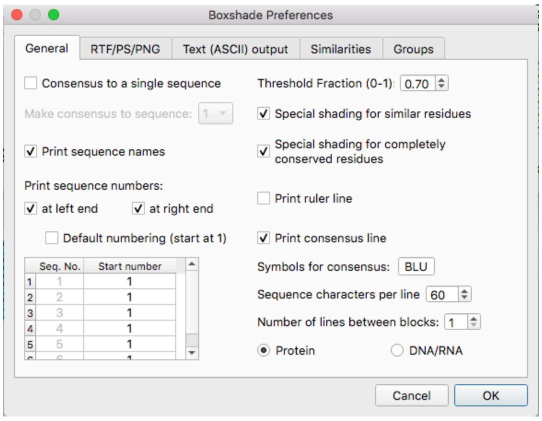

vprotends.fasta

```
Hpivt2    1  RERArGNHRREWSiAWVGDQvKvFEWCNPRCAPvTASARkFTCT
SV5       1  RDTG-GFHRREYSiGWVGDEvKvTEWCNPSCsPiTAAARrFECT
PDV       1  CEPIkKGHRREVSlTWNDDRCWiDKWCNPICtQvNWGVIrAKCI
CDV       1  CGSIkKGHRREVSlTWNGDSCWiDKWCNPICtQvNWGIIrAKCF
Measles   1  GTPIkKGHRREISlIWNGDRvFiDRWCNPMCsKvTLGTIrARCT
RPV       1  EKPIkKGHRREIDlIWNDGRvFiDRWCNPTCsKvTVGTVrAKCI
Mumps     1  PDNPrGGHRREWSlSWVQGEvRvFEWCNPICsPiTAAARFHSCK
Hpivt4    1  PTPR---HRREYSiSWVNGRTTiSEWCNPCCAPvKSTASVEKCT
Bpivt3    1  GWKGkRGHRREHSiYREGDYiITESWCNPICsKiRPVPRQESCV
Sendai    1  SDSTkKGHRREHIiYERDGYiVDESWCNPVCsRiRVIPRrELCV

Hpivt2   51  ICGECEGD-H----------------  59
SV5      50  TCSECERD-T----------------  58
PDV      51  VCDDCKDD-PEMQNRIWYETPTRETK  75
CDV      51  TCNECKDD-PEMQTRVWHATPSQDLK  75
Measles  51  VCEQCRTD-TGVDTRIWYHNLPEIPE  75
RPV      51  VCEQCITD-SGIENRIWYHNLADIPE  75
Mumps    51  KCDQCERD-YGPP-------------  62
Hpivt4   48  ICELCIRD-P----------------  56
Bpivt3   51  QCGYCIEDRLSSRSNDWSINGS----  72
```

from this window the image may be saved by pressing the PNG icon (program will ask for the name of the file to save using the normal dialogs).

Open a dialog to adjust any of the program parameters. The Preferences dialog has a number of tabs:

1) General: This pane controls many parameters controlling what will be included in the image and what kind of consensus will be used.


Boxshade Preferences

□ Consensus to a single sequence

Make consensus to sequence: 1

☑ Print sequence names

Print sequence numbers:

☑ at left end ☑ at right end

□ Default numbering (start at 1)

	Seq. No.	Start number
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1

Threshold Fraction (0-1): 0.70

☑ Special shading for similar residues

☑ Special shading for completely conserved residues

□ Print ruler line

☑ Print consensus line

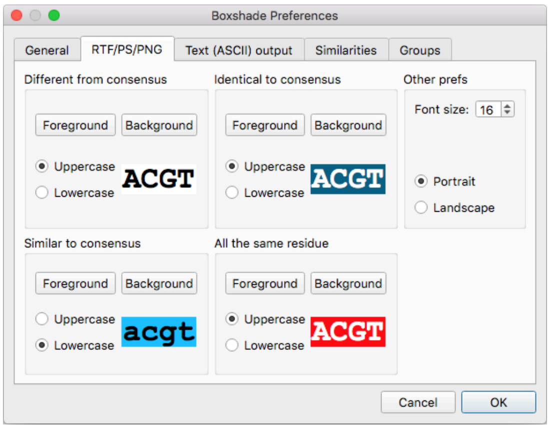
Symbols for consensus: BLU

Sequence characters per line 60

Number of lines between blocks: 1

● Protein ○ DNA/RNA

2) RTF/PS/PNG : Foreground and background colours are set here, as also whether the amino acid/base will be shown in uppercase or lowercase for a particular type of conserved residue, what font size to use, and Portrait/Landscape for PS files.


Boxshade Preferences

Different from consensus

● Uppercase ○ Lowercase ACGT

Identical to consensus

● Uppercase ○ Lowercase ACGT

Other prefs

Font size: 16

● Portrait ○ Landscape

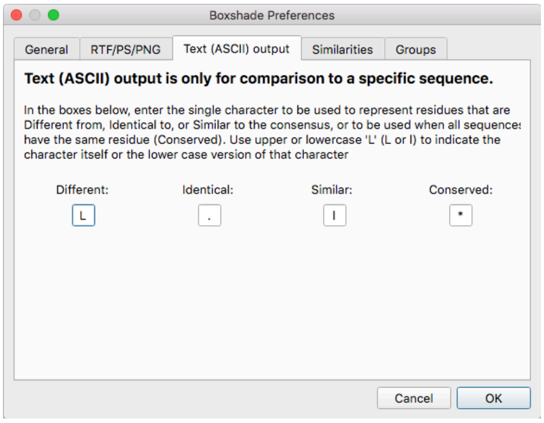
Similar to consensus

○ Uppercase ● Lowercase acgt

All the same residue

● Uppercase ○ Lowercase ACGT

3) ASCII/Text: This is a very simple dialog that allows the user to set the character that will be used for amino acids/bases that are different to/identical to/similar to/globally conserved/ relative to the consensus sequence.

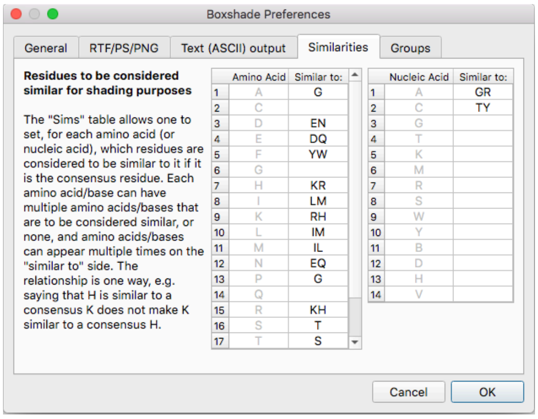


Text (ASCII) output is only for comparison to a specific sequence.

In the boxes below, enter the single character to be used to represent residues that are Different from, Identical to, or Similar to the consensus, or to be used when all sequence: have the same residue (Conserved). Use upper or lowercase 'L' (L or l) to indicate the character itself or the lower case version of that character

Different:	Identical:	Similar:	Conserved:
L	.	l	*

4) Sims: Table to allow the user to define which amino acids/bases will be considered similar to each amino acid/base in the consensus.



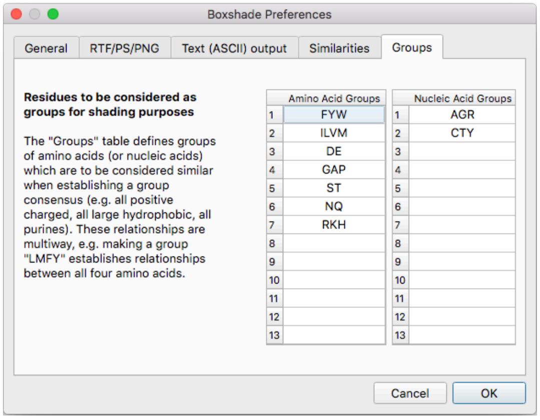
Residues to be considered similar for shading purposes

The "Sims" table allows one to set, for each amino acid (or nucleic acid), which residues are considered to be similar to it if it is the consensus residue. Each amino acid/base can have multiple amino acids/bases that are to be considered similar, or none, and amino acids/bases can appear multiple times on the "similar to" side. The relationship is one way, e.g. saying that H is similar to a consensus K does not make K similar to a consensus H.

	Amino Acid	Similar to:
1	A	G
2	C	
3	D	EN
4	E	DQ
5	F	YW
6	G	
7	H	KR
8	I	LM
9	K	RH
10	L	IM
11	M	IL
12	N	EQ
13	P	G
14	Q	
15	R	KH
16	S	T
17	T	S

	Nucleic Acid	Similar to:
1	A	GR
2	C	TY
3	G	
4	T	
5	K	
6	M	
7	R	
8	S	
9	W	
10	Y	
11	B	
12	D	
13	H	
14	V	

5) Grps: Table to allow the user to define which amino acids/bases will be considered similar when forming a “consensus by similarity” (see above).



Residues to be considered as groups for shading purposes

The "Groups" table defines groups of amino acids (or nucleic acids) which are to be considered similar when establishing a group consensus (e.g. all positive charged, all large hydrophobic, all purines). These relationships are multiway, e.g. making a group "LMFY" establishes relationships between all four amino acids.

	Amino Acid Groups
1	FYW
2	ILVM
3	DE
4	GAP
5	ST
6	NQ
7	RKH
8	
9	
10	
11	
12	
13	

	Nucleic Acid Groups
1	AGR
2	CTY
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	

A note on fonts

The PostScript (PS) output expects Courier-Bold to be available on the system where the PS file is viewed; this font was chosen because it is a traditional PS font that is normally available as a PS or TrueType font on MacOS X and Windows. On Windows systems, if using Ghostscript or similar to view a PS file, a similar-looking font is used (Nimbus Mono). The image may look strange if no Courier (or substitute) font is available on your system (e.g. systems set up using non-Roman scripts such as Chinese or Japanese).

The font used in the RTF files is Courier New by default. This is primarily historical, because it was assumed that the files would be opened with MS Word. An advantage of the RTF files, however, is that one can change the font just as with any word processed document; note that if the font used is not a monospaced font the alignment will no longer be properly aligned.

The font used in the PNG pictures depends on the operating system. I tried to avoid any system dependence, but the fact remains that Courier on MacOS X looks a lot better than Courier New, while the reverse is true on Windows, so on those two platforms the better font is selected. On Linux, it is very hard to predict what fonts will be available. On a basic Ubuntu installation, you are likely to get the font Liberation Sans Mono, which is OK-ish. If you install the free MS-Core TrueType fonts, you will get Courier New, which is better.

Acknowledgements

Thanks to:
Kay Hofmann for the original BOXSHADE.
The people at River Computing who make the PyQt bindings between python and the Qt application framework
The people at the Qt Company who make the Qt application framework
Freepik at www.flaticon.com for some of the icons I use