Unix, Perl and BioPerl

Session 2: Sequence analysis with Perl (introduction)

Exercise 1: Retrieving and aligning a list of human-mouse orthologs

Goal: Learn the some common Perl commands and file handling while generating sequence files and aligning human-mouse pairs of orthologous genes.

See `http://jura.wi.mit.edu/bio/education/bioinfo2005/unix-perl/` for course page

All shell (i.e., non-Perl) commands must be enclosed in back ticks and end with a semicolon (ex: `` `ls`; ``) Many of these utility commands (cat, grep, etc.) exist as comparable Perl commands, but using the shell commands should be quicker to figure out for this exercise.

To do:

- Log onto hebrides and enter the 'unix_class' directory. Create a directory called `perl_1` and enter it.

- Copy starting script and data file from /home/george/perl_1 to your current directory.
  script = `align.pl`
  data file = `human_mouse_pairs.txt`

- Check permissions of script and change to 744 if necessary.
  `chmod 744 align.pl`

- Open `align.pl` with pico (command line), nedit (Xwindows), or another text editor. It may be helpful to look at `human_mouse_pairs.txt` too (using a text editor or the `more` command)

- Under the line "# 0":
  Define human ($humanAcc) and mouse ($mouseAcc) accessions.
  Hint: the first element in an array is the 0^{th} element.

- Run the script, check the output, and debug.

- Under the lines "# 1" and "# 2":
  Fastacmd extracts a sequence from a BLAST-formatted database. Syntax for fastacmd is

  `fastacmd -d database -s accession`
  where database in this case is "nr", the default GenBank protein database.
  Redirect output to appropriate file.

- Note: Most alignment outputs truncate the sequence headers, and sometimes they're so truncated that they're useless (i.e., if the sequence is identified by the first word in the header, which is "gi"). To prevent that, the Unix "sed" command is used to remove the first part of the header, only leaving what comes after the "ref|". The output file is then renamed to the original file.

- Under line "# 3": Some alignment applications (clustal) require all sequences in the same file, so this command will generate that file.

- Under line "#4": Choose a type of alignment to perform and write the appropriate command

Optimal global alignment (Needleman-Wunsch algorithm):
`needle` is the command in the EMBOSS package
brief help: `needle -help`
better help: http://www.hgmp.mrc.ac.uk/Software/EMBOSS/Apps/needle.html
basic syntax:
```
needle seq1 seq2 -outfile nameOfOutFile -auto
```

Optimal local alignment (Smith-Waterman algorithm):
`water` is the command in the EMBOSS package
brief help: `water -help`
better help: http://www.hgmp.mrc.ac.uk/Software/EMBOSS/Apps/water.html
basic syntax:
```
water seq1 seq2 -outfile nameOfOutFile -auto
```

Clustalw:
a popular multiple alignment program (not ideal for only 2 sequences)
the command-line version of ClustalX (also on hebrides)
for help, `clustalw -help`
basic syntax:
```
clustalw -INFILE=seqFile -TYPE=DNA -OUTFILE=nameOfOutFile
```
where you can specify seqFile and nameOfOutFile

- 5: Generate one big file of all alignments, including each pair of sequence headers and the alignment itself.

  To get the sequence headers, use the Unix "grep" command.
  To append the current alignment to the end of the big alignment file, use the Unix "cat" command.

- One possible solution (a completed script) will be in /home/george/solutions