Unix, Perl, and Python

Session 2: Perl for Bioinformatics

Exercise 2: Retrieving and aligning a list of human-mouse orthologs

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

The script you'll write will take as input a list of human-mouse sequence IDs. The gene sequence of each human and corresponding mouse ortholog is extracted from a BLAST-formatted database. The two sequences can then be aligned locally or globally. In this way you can automate pairwise alignments of any number of sequences. As each alignment file is created, it's concatenated to the end of the previous alignments, so at the end you have one big alignment file.

See **`http://jura.wi.mit.edu/bio/education/hot_topics/Unix_Perl_Python/`**
for the 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:

1	Log onto tak, which should take you to your home directory.
2	Create a directory (if you haven't already) called `perl_class` and enter it.
3	Copy starting script and data file from `/nfs/BaRC_Public/Unix_Perl_Python/Perl/` to your current directory. script = `align.pl` data file = `human_mouse_pairs.txt`
4	Check permissions of script and change to executable if necessary. `chmod +x align.pl`
5	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). The first field is the human accession, and the second field is the mouse accession.
6	Under the line "# 0": Define human ($humanAcc) and mouse ($mouseAcc) accessions. Hint: the first element in an array is the 0th element.

7	Run the script `./align.pl` check the output, and debug if necessary. You can expect to get some “No such file or directory.” warnings because the starting script is expecting some files that you haven’t yet created.
8	Under the lines “# 1” and “# 2”: Write commands to extract each sequence from a BLAST-formatted database with *fastacmd*. The syntax for fastacmd is `fastacmd -d database -s accession > outputFile` where database in this case should be “nr”, the default GenBank protein database. Output should be redirected to the appropriate file (in place of outputFile). When you run the script now, you shouldn’t get any warnings.
9	Note *a*: For fastacmd, blastall, and other applications in the BLAST suite, you need to show the path to the location of the database. An exception is when the database is in a directory defined as the “BLASTDB” by the environment on the computer and/or your account. On the command line (not within Perl), try the command `echo $BLASTDB` and you’ll see where the ‘nr’ BLAST database is located.
10	Note *b*: 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, we use a perl command (within our perl script) to remove the first part of the header, only leaving what comes after the “ref|”.
11	Under line “# 3”: Some alignment applications (clustal) require all sequences in the same file, and the Unix “cat” command can be used to generate that file.
12	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://emboss.open-bio.org/wiki/Appdoc:Needle 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://emboss.open-bio.org/wiki/Appdoc:Water 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
13	Under line "#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. Run the script and take a look at alignment-all.txt Does alignment-all.txt include your desired output?
14	One possible solution (a completed script) is in `/nfs/BaRC_Public/Unix_Perl_Python/Perl/solutions/` The solution script includes all three alignment methods, each written in a subroutine (like `sub doClustal { }`). When Perl calls a subroutine using a command like `doClustal()`, all commands in that subroutine are executed. Using a subroutine is a way of placing a set of commands for one task in one place (to help organize your code). If you had these commands in the main part of the script (under #4 and #5), they would be executed in the same way.