## An introduction to running BLAST on the command line.

**Preface:** Often it is useful to use BLAST at the command line, either through an API (i.e., still using a remote database like NCBI, but doing things in bulk), on your own computer, or using a remote computer. One reason to do this is to run a lot of sequence in bulk. Another is if the information you are looking for is not in the database (i.e., if it is brand new). Third, your queries will often run faster on a nice cluster than at NCBI (they have great speed, but a lot of demand). And there are various other reasons where it is useful to have fully customizable BLAST. We have such a setup at Earlham and this is what you will use for this lab. GUIs are a great way to get started with many programs but ultimately the power of computation much more accessible from the shell.

**Part I:** Log into the clusters and the server you will use

* The local servers all use Bash for their shell, all the commands you learned in the first lab work there
* Open a terminal/shell on your laptop
* $ ssh <cluster\_username>@cluster.earlham.edu # you received this via email recently  
  The first time you will have to accept the security key for the server
* Use ssh to connect to your assigned server ($ ssh <hostname>) - we will all use 'whedon'
* All the servers share the same filesystem, you can start your work on one machine and then finish it later on another
* The command to move files back and forth between your laptop and the clusters is scp (secure copy).
* Make a directory on the cluster called 'bioi', and within that make another directory called 'blast'

A typical command to move files from my laptop to my bioi/homework directory on the clusters is:  
$ scp file.dat <cluster\_username>@cluster.earlham.edu:bioi/homework   
  
A typical command to move files from my bioi/homework directory on the clusters to my laptop is:  
$ scp <cluster\_username>@cluster.earlham.edu:bioi/homework/file.dat file.dat  
  
A typical command to move all the files from my bioi/homework directory on the clusters to my laptop is:  
$ scp <cluster\_username>@cluster.earlham.edu:bioi/homework/\* .

**Part II:** Get and organize data

* Make sub-directories (folders) within your blast directory that using the following names: “genomes”, “transcriptomes”, “proteomes”, “annotations”, “blastoutput”, “queries”
  + I think it is pretty clear what will happen in each of these, but just in case:
    - You will download genomes, transcriptomes and proteomes to those directories and they will be your databases.
    - You will put your query sequences in that folder.
    - You will direct your outputs to the output folder.
    - Trust me when I say that organization really matters here!
* Google “NCBI *Drosophila melanogaster* genome” and select/enter the best hit URL or go to NCBI and in the top search bar select “genomes” and then type in *Drosophila melanogaster*. Either way should get you to NCBI’s genome page for the model organism Fruit Fly (*Drosophila melanogaster*, note that this is formally in ital., but the computer doesn’t care for these searches).
  + There is a box at the top with the following text:
    - “Download sequences in FASTA format for **genome**, **transcript**, **protein”**
    - Download these files into their appropriate folder that you made above. I advise using the -O option (look up what that does). You can “copy link address” by ctrl-clicking on the link.

**Part III:** Unpack the data and make BLAST directories.

* In each of the folders with data, make a BLAST directory. This is mostly the FASTA and another file (.fai) that will have coordinates for each base in the FASTA file.
* First, you must unzip the file. You can use the shell command “gunzip” (which is because the file is g-zipped)
* \*\*\*make sure your blast module is loaded, type: module load BLAST+/2.3.0
* Second, make the BLAST directory using the syntax below
  + $ makeblastdb -in FILENAME -parse\_seqids -dbtype NUCL/PROT
  + In the above:
    - Substitute your actual file name for FILENAME
    - Use in lowercase nucl for nucleic acid sequence and prot for amino acids

**Part IV:** Find what to look for (i.e., what is your query?).

* The Hox genes were a major discovery made in fruit flies that resulted in the awarding of a Nobel prize and the revolution of our understanding of animal (and plant) development.
* Search for the following gene ID in NCBI: **Dmel\_CG11648** . Get the protein fasta and save it to a file in your “queries” directory (name the file something logical).
  + You can do this in varying ways, I give you 4 below:
    - Download it from the right-click menu on the gene model.
    - Go to the protein accession for the aa product of the above gene (it starts with NP) and then copy and paste the fasta into a text file (for example, in nano, on the cluster).
    - As above, but save a local copy and then scp the file from your computer to the cluster.
    - Use a web api call (below, note this is one line, but gets wrapped):
      * curl "https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=protein&id=NP\_996220.1&rettype=fasta&retmode=text" > NP\_996220.1.faa
      * Note: an API (application programming interface) is a way to access someone else's data and code from a remote instance.
      * Note2: a regular curl retrieves html garbage...try it.
      * Note3: there are a lot of options on how you get your sequence, these are all of the things following 'efetch.fcgi?'
* Now BLASTp this sequence into the proteome file using the syntax below.
* $ TYPE -db DATABASEFILE -query QUERYFILE -out OUTPUTFILENAME -outfmt 6
  + TYPE = which BLAST type are you doing, i.e., BLASTn, BLASTp, etc. (note: this must be typed lowercase)
  + DATABASEFILE = the name of the fasta file that you are searching (and that you have set-up for BLAST in Part 2.
  + QUERYFILE = the FASTA file that contains the sequence you are searching for.
  + OUTPUTFILENAME = what you want to call the file that the output will contain.
  + Outfmt options - these are listed at the end of this doc, a delimited file is often very useful.
  + There are many other options you can use. You can get a long list of options (some of which will be very handy later) by typing “blastn -help”
* **Describe your best hit. Does this make sense?**

The best hit for

is as followed:

Query Hit

NP\_996220.1 NP\_524896.2 100.000 270 0 0 1 270 224 493 0.0 575

* Repeat this, but blast this sequence into the genome and transcriptome.
* **How do these differ from each other and how do they differ from the proteome hit?**

Part V: Let’s use the above IDs to retrieve sequences and then BLAST a lot of stuff at once.

* There are tools that allow for the manipulation of genomic files. One set is SAMtools (SAM is for Sequence Alignment/Map - note, SAM is the uncompressed version of BAM, which we saw earlier).
  + What we want is to be able to provide a list of IDs and then retrieve the sequences in FASTA format. Below is syntax for such a tool.
  + \*\*\*make sure your blast module is loaded, type: module load SAMtools/1.4
  + $ xargs samtools faidx FILETOSEARCH < FILEWITHIDS > OUTPUTFILE
  + First try to extract the FASTA sequence of NP\_524896.2 (make a new file using nano and then put that ID in there - next feed that file name into the command above (xargs samtools…).
    - Your result should be a file that has that ID followed by its protein sequence.
* We can now take this to the next level. What if we want to extract BLAST results and turn them into sequence. We need the IDs - these are in BLAST table output.
* Below is a sample BLAST output. From this my matches for PO2833.1 are in the second column (NP…). So, I want to cut out only this second column and then redirect that into a new file.
* I could use something like the below to do this in bash:
  + $cut -f 2 antp.out > antp\_names.txt
    - In the above: 'cut -f' cuts out the second field.

P02833.1 NP\_996175.1 100.000 378 0 0 1 378 1 378 0.0 763

P02833.1 NP\_996170.1 100.000 378 0 0 1 378 1 378 0.0 763

P02833.1 NP\_996168.1 100.000 378 0 0 1 378 1 378 0.0 763

P02833.1 NP\_996167.1 100.000 378 0 0 1 378 1 378 0.0 763

P02833.1 NP\_996174.1 98.942 378 0 1 1 378 1 374 0.0 748

* In your above BLASTp search (from part IV), extract the IDs that you hit and then extract their protein sequence.
* Next BLAST these into the transcriptome and extract those IDs.
* **Briefly, describe your output.**
* Final Product:
  + Write this into a shell script (with descriptive annotations).
    - The script should do the BLASTp from part IV, cut, extract, do the BLAST of those into the transcriptome, cut and extract. The final resulting file should be FASTA file of your transcripts.
      * NOTE:The script should not assume that you have the data or the database.
      * NOTE2:The BLAST part of the script should specify to keep **only the best** BLAST hit (i.e., there is only one target sequence). Otherwise this will take really long to run!

The following is from: https://www.ncbi.nlm.nih.gov/books/NBK279684/

\*\*\* Formatting options  
 -outfmt <String>  
 alignment view options:  
 0 = pairwise,  
 1 = query-anchored showing identities,  
 2 = query-anchored no identities,  
 3 = flat query-anchored, show identities,  
 4 = flat query-anchored, no identities,  
 5 = XML Blast output,  
 6 = tabular,  
 7 = tabular with comment lines,  
 8 = Text ASN.1,  
 9 = Binary ASN.1,  
 10 = Comma-separated values,  
 11 = BLAST archive format (ASN.1)