Assignment 8 _2024: Comparing divergent genomes using blastp

Your name:
Your email address:

Comparing divergent genomes using blastp from the command line

For today's first exercise you will use protein sequences (.faa files) from two completely sequenced genomes as multiple fasta formatted files. Possible are one bacterium, one archaeon or one Eukaryote. Any combination of organisms from the different "domains of life" should work nicely. The lab8.zip archive includes a few examples, but please go after organisms that you are interested in.

Follow the instructions from lab6 to download additional genomes

The genome download page at NCBI currently only works on **Chrome and Firefox**. Go to the NCBI's genome page and search for a bacterial, archaeal or eukaryotic species you are interested to explore (e.g., Rhizobium sp. and Phaseolus vulgaris).

Which species did you select? Your answer:

Are there complete genomes listed? (if not, use the genome with the largest number of protein coding genes. If the table does not list any protein coding genes, pick another genomes.) Your answer:

How many protein coding genes are in the genomes of choice? Your answer:

In the Genome Table (remember Safari **does not work** for the download), place a checkmark into the row of a suitable genome, then click on download, then select download package.

In the window that opens, place checkmarks into Refseq only (if there is a Refseq genome), and Protein (FASTA).
Then click Download. A compressed file should appear in your download folder. Unzip this NCBI_dataset folder.

Move to the ncbi_dataset folder, move to the ncbi_dataset folder, open the data folder, open the genome folder. Rename the .faa file into something that lets you know from which organism it is (e.g. Rhizobium_sp.faa or Phaseolus_vulgaris.faa) and move the files into your lab8 folder.

Open a filezilla connection to Xanadu (see **here** how to do this), create a lab8 subdirectory, and copy the contents of you local lab8 folder to the lab8 folder on Xanadu.

Establish an ssh connection to your account on Xanadu

`cd lab8` change to the lab8 directory

`ls` to check that the perl scripts and gneome files are present

move to a compute node:
`srun --pty -p mcbstudent --qos=mcbstudent --mem=2G bash`
if you have a very large genome, use **`--mem=5G`**

Check the beginning of your *.faa files:
`head *.faa`

Check the available modules – in case of blast you need to use the version number.
`module avail`

Load the modules for blast and perl:
`module load blast/2.13.0`
`module load perl`

Pick one of your "genomes" to turn into a searchable databank (in this example Borreliella_burgdorferi.faa):
```
makeblastdb -in Borreliella_burgdorferi.faa -dbtype prot -parse_seqids
```
or
```
makeblastdb -in Saccharomyces_cerevisiae_S288C.faa -dbtype prot -parse_seqids
```
or
```
makeblastdb -in Homo_sapiens.faa -dbtype prot -parse_seqids
```

This should be pretty fast. A report on how many sequences were added to the databank is displayed.
To search the databank we will use one "genome" as "query", and the databank created above as databank.

Note#1: to retrieve the high scoring pairs of sequences, it will be helpful to also convert the sequences from you query genome into a databank
Note#2: Things can become confusing, especially when things go wrong. Therefore, take good notes, including the commands you executed ...

In the example below, the databank is Borreliella_burgdorferi.faa and the genome from Haloferax is the query (in general, it is preferrable to using the smaller genome as query, else in the eukaryotic genome you get hits with all the paralogs to the same gene in the bacterial/archaeal genome):

```
blastp -query Haloferax_volcanii_DS2.faa -db Borreliella_burgdorferi.faa -out
blast_all.txt -outfmt 6 -evalue 1
```
or
```
blastp -query Borreliella_burgdorferi.faa -db
Saccharomyces_cerevisiae_S288C.faa -out blast_all.txt -outfmt 6 -evalue 1
```
or ...

(the first is fast, the 2nd will take 5 minutes, a large genome like the one from Phaseolus more than 20 minutes)
Depending on the size of the databank and the size of the query this might take a few minutes to finish. If you want your command-line prompt back, type `ctrl z` to halt the process, and then enter `bg` to continue the search in background. `ps` gives you a list of things that are still running.

You can check the progress in filezilla watching the size increase in the output file.

-outfmt 6 specifies a tabular output format. (We use an e-value of 1 to also obtain some insignificant hits.)

The blast program will now take every sequence in query file and do a blast search against the database (i.e., this performs a couple of thousand blast searches).
This will take a few minutes. While waiting check in what other students are doing :) Or check this listing of options to use with the blast command.

Which option turns the filter for low complexity on/off?
your answer:

We are also interested in the best blast hit for each query. The script blastTopHit.pl goes to thought he output table and for every query only prints out the first entry.
(It is possible to restrict the blast search to return only one hit, but it is alleged that this only return the first significant hit encountered, not the best one).
The perl script should be already in the lab8 folder:

To run the perl script, and providing that you did not change the output file in the blast command, execute:
```
perl blastTopHit.pl blast_all.txt > blast_top.txt
```

Check the files blast_all.txt and blast_top.txt using more or head:
```
head blast_all.txt blast_top.txt
```

Here is a description of the columns in the outfmt -6 option. A file containing a table header is in the lab8 folder called header.txt (check it contents with **`cat`** or **`head`**)

Use the cat command to add the header to the blast output files:

```
cat header.txt blast_top.txt > blast.top_h.txt
```

Check the files with header blast_all_h.txt and blast_top_h.txt using more or head:

```
head blast.top_h.txt
head blast.all_h.txt
```

Go back to **Filezilla**. Navigate to your lab8 directory, and transfer the blast.top_h.txt file to your Desktop. Load it into Excel. On my computer I can right click on the file and select open with and then Excel. On the Macs in the lab this maybe be slightly more complicated, but you need to use the MSWord application, not the web 365 version.

Click in the top left field of the table, then convert the spreadsheet into a table with headers.

This will allow you to sort the table on the different columns.

What is the accession ID pair of the most conserved protein between your two genomes (sort the table first on bitscores (higher to lower) , **then** on e-values (lower to higher)?

To learn what the most similar sequence(s) encode, we can copy the Accession numbers for the most similar matches (use the database IDs in the second column), paste them into a text editor (I use BBedit), replace the new line symbols "\n" with "," and copy the comma separated list into the following command:

```
blastdbcmd -entry IDnumber1,IDnumber2,IDnumber3 -db YourDatabaseName
```

for example (all on one line):

```
blastdbcmd -entry
WP_002662068.1,WP_106011477.1,WP_002557027.1,WP_002557108.1,WP_1
06013134.1,WP_002656192.1,WP_106011525.1,WP_002665762.1,WP_00265
7046.1,WP_106013121.1,WP_002661984.1,WP_010883927.1,WP_002657221
.1,WP_002661428.1,WP_106011512.1,WP_002657221.1,WP_106011512.1,W
P_002661428.1,WP_002656799.1,WP_002661757.1,WP_106011512.1,WP_00
2660520.1,WP_002661852.1,WP_002656738.1,WP_002660520.1,WP_002660
666.1,WP_106011471.1 -db Borreliella_burgdorferi.faa
```

This should return the fasta formated sequences of the matches.

What is given as function in the fasta annotation lines?

In Excel, select the third column from the left (this is the percent identity for each BLASTp match), and Insert -- Statistic Chart (all-blue column chart icon in the Charts section) -- Histogram
(Aside: I had problems in editing the histograms this creates on a Mac. It worked much better, when I selected tools, analysis tool pack (you might need to install this first), Histogram. And I added a column somewhere that gave the bins from 0 to 100% in 2% increments)

(More about histograms in Excel, including the formula used for default binning.)

You get a histogram of all percent identities. Repeat the histogram for E-values ≤ 10^{-3}, and E-values >.001

Do you observe a smooth distribution of % identity values (a distribution with two peaks would be noteworthy)?

What can explain the difference (or lack thereof) in the histograms for significant and insignificant hits?

Can you explain, why % identity is not a good criterion to distinguish significant from insignificant hits?

For the following questions, the BLAST Command Line Applications User Manual may be helpful → link

How can you get information on the possible parameters in blastp using the command line? (try blastp -h first)

How can you set the wordsize to 2?

How can you filter the query sequence for regions with low complexity?

Can you think of a better approach turn sequence identity into a useful measure? Try it out. Do you get better results? (Hint: You could add a column to your spreadsheet giving % identity times alignment length)

Do not forget to email the completed form, and to put the histograms and other results into your notebook!