BF527 Lab 8

Basic Local Alignment Search Tool (BLAST)

BLAST is a multipurpose bioinformatics algorithm for searching a query sequence against a database of other sequences. The algorithm returns local alignments of varying significance. Today you will try both the on- and off-line versions of BLAST. Your goal will be to identify a protein sequence and then scan its host proteome for duplicated genes (or other interesting alignments). You will not be coding in python today, but you will be asked to think about how python scripting might be useful in these applications.

While walking around in the lab yesterday I noticed an mRNA transcript stuck to the bottom of my shoe. I promptly amplified the sample and determined its sequence, which corresponded to the following translated protein:

>Unknown putative shoe-interacting protein 1
MPKKSIEEWEEDAIESVPYLASDEKGSNYKEATQIPLNLKQSEIENHPTVKPWVHFVAGGIGGMAGAVVTCPFDLVKTRLQSDIFLKAYKSQAVNISKGSTRPKSINYVIQAGTHFKETLGIIGNV`

Part A: Online BLAST

Determine the identity of this protein using the blastp tool on the BLAST website.

From which genome is it most likely to have originated?

Saccharomyces cerevisiae

Perform a second web BLAST, this time restricting the search database to the genome identified in Step A.

Are there any duplicated copies of this protein in its host genome?

If so, are the reported functions of the duplicates similar to those of the original protein?

How do you explain the other significant alignments observed (if any)?

Yes, both Rim2p and other two NAD+ transporter have relatively high e-value. And their reported functions are similar. Other significant alignments are also located in the mitochondria of Saccharomyces cerevisiae with similar tranport functions.

Part B: Local BLAST

Preliminaries

This task is done on a server! Before completing it, you must first do the following:

1. Download the files `query.fasta` and `proteome.fasta` from Blackboard to your computer
2. Transfer these files to scc using your favorite file transfer program. MobaXterm supports SFTP sessions. SFTP is secure file trasfer protocol. If you don't have a favorite program, FileZilla or cyberduck or Fetch are all good solutions:
   A. Click on "Open Connection"
   B. Choose "SFTP" from the dropdown list
   C. Enter `scc2.bu.edu` as the server name
   D. Use your BU login name and password
   E. Make sure port 22 is selected (for secure FTP).
   F. Once you are connected, upload `query.fasta` and `proteome.fasta` by clicking "upload"

OR simply use

```
scp file.name username@scc2.bu.edu:/projectnb/bf527/username/
```

from MobaXterm or Mac terminal to transfer the files to the cluster.

UNIX TIP: When working on the command line, you can redirect information normally printed to the screen into a file by following your command with a “>” and a file name:

```
./program_name argument1 argument2 etc. > output_file.txt
```

You can also redirect information normally printed to the screen into another program by following your command with a “|” and a program name. For example, to scroll through your program’s output:

```
./program_name argument1 argument2 etc. | less
```

Running local BLAST

To avoid the time delays of sending information over the internet and waiting in line for access to NCBI’s servers, it is often convenient to run BLAST completely offline. Typically this is done when one wishes to search a reduced database. BLAST is preinstalled on SCC. To make the BLAST module available load it.

```
module load blast+/2.7.1
```

To see other available versions:

```
module avail blast
```

Before you can use BLAST to search for your sequence, you must format your database (originally in FASTA format) for fast searching by BLAST. This is done with the program `makeblastdb`. In order to learn more about `makeblastdb`, type the following in the command line:

```
makeblastdb -h
```

In order to format the database you need to search, type:

```
makeblastdb -in proteome.fasta -dbtype prot -parse_seqids
```

The `-parse_seqids` flag parses the IDs of the input FASTA file, and makes the output database searchable by those IDs.

What does `-dbtype prot` indicate about our input file?

database type is a protein sequence

You can now search against this database using a variety of BLAST programs. Here we'll use `psiblast`. To learn more about `psiblast`, type the following in the command line:

```
psiblast -h
```

Which arguments are needed to search the query sequence (query.fasta) against your newly formatted database (proteome.fasta)?

Run your program on the command line.

psiblast -query query.fasta -db proteome.fasta

PYTHON TIP: One advantage of using BLAST on the command line is that you can access it from your Python scripts. Imagine that instead of searching for the duplicates of one gene in your genome, you instead wished to search for duplicates of *all genes* in the genome (repeating Step B once for each gene). You would not want to do this manually. The python library `os` provides a method `system()` that allows you to make system calls (i.e. run commands in the Unix shell) from within python. For example:

```
import os
os.system("ls *.py > MyScripts.txt")
```

This Python code will call Unix to create a list of your Python scripts and then pipe them (using ">") into a file called "MyScripts.txt." Note that you could use the same procedure to execute the BLAST commands that you discovered earlier.
