CB2-101: Sequence Similarity Search

Malay (malay@uab.edu)

Nov 11, 2016

Contents

1 Simple BLAST

BLAST p53 human sequence against SwissProt database.

1.1 Download sequence files

Download SwissProt FASTA file file:

```
wget ftp://ftp.uniprot.org/pub/databases/uniprot/knowledgebase/uniprot_sprot.fasta.gz
```

Find the id of P53 using the Uniprot API.

```
wget http://www.uniprot.org/uniprot/?query=organism:9606+AND+gene:P53&format=tab&columns=id
```

The first id is `P04637` is the actual id of `P53_HUMAN`.

1.2 Exercize

Write a script to extract sequence by id from a FASTA file and use it to extract p53 sequence from SwissProt FASTA file.

1.3 Create the BLAST datbase and run BLASTP

```
formatdb -i uniprot_sprot.fasta
blastall -p blastp -i uniprot_sprot.fasta -o output.bla
```

1.4 Download blast

BLAST has new C++ version called `blast+`. But we will stick with the older version.

Download the BLAST executable from NCBI FTP site:

ftp://ftp.ncbi.nlm.nih.gov/blast/executables/release/2.2.26/

Edit your `~/.bashrc` to put the executable on your path. You also may need to create a `.ncbirc` depending of the version of the BLAST. For version 2.2.26 you really do not need to create one.

2 Advance BLAST

Homologs are sequences that are related by descent. **Orthologs** are homologs present in two separate species. This does not assume that two homologs are similar in sequence. However, it's a common practice to find orthologs using sequence similarity.

One of the simple ways to find orthologs of a protein from one species in another is to find the protein's **best hit** in the second species and consider the hit as ortholog only when the the hit itself also has the starting protein as the best hit. This relationship is called "Reciprocal Best BLAST hit" or RBH.

We will use RBH to determine all the orthologs between *Saccharomyces cerevisiae* and *Schizosaccharomyces pombe*. This requires all proteins in this two genomes to be searched against all other proteins. This type of BLAST searches are commonly called **all vs all** BLAST.

2.1 Downloading and preparing the input files

First let's create the SC proteome file.

```
# Create a directory for SC
mkdir -p SC

# Change to that directory
cd SC

# Download all the files of SC from NCBI
wget --quiet ftp://ftp.ncbi.nih.gov/genomes/Fungi/Saccharomyces_cerevisiae_uid128/*.faa

# Combine all the .faa files into one file
cat *.faa >sc.fas

# We no longer need the .faa files
rm -rf *.faa

# Compress the file
bzip2 -9 sc.fas
```

Then we will create the SP proteome file

```
mkdir -p SP
cd SP
wget --quiet ftp://ftp.ncbi.nih.gov/genomes/Fungi/Schizosaccharomyces_pombe_uid127/*.faa
cat *.faa >sp.fas
rm -rf *.faa
bzip2 -9 sp.fas
```

2.2 Formatting the BLAST database

For searching a sequence against a database, BLAST requires the database to be formatted in a specific way. There are some pre-formatted databases available from the FTP site:

ftp://ftp.ncbi.nlm.nih.gov/blast/db/

However, in our case, we need to format our own database. First we will combine the two genomes into one file.

```
bzcat *.bz2 | bzip2 -c >all.fas.bz2
```

The software that formats a fasta file searchable by BLAST is called `formatdb`. You can view the full list of options by running:

```
formatdb -
```

You can simply run `formatdb` using the following command

```
formatdb -i <inpufile>
```

This will create a datbase with the same name as the input file. However, this requires a unzipped flat file, which is a no no in our case. You can run formatdb using the following command:

```
bzcat all.fas.bz2 | formatdb -i stdin -o T -n "all"
```

The `-o` options tells `formatdb` to create an index of the sequences. This will come handy if we would like to get a sequence out of this database later. `-n` option is necessary, because when reading from `stdin`, `formatdb` has no idea what to call the database.

Depending on your computer ram, you might have to change `-v` to fit the database entirely in the RAM. For a list of tips and tricks read the formatdb documentation:

ftp://ftp.ncbi.nlm.nih.gov/blast/documents/formatdb.html

2.3 `fastacmd`

`fastacmd` is a very useful utility supplied with BLAST. For instruction see:

ftp://ftp.ncbi.nlm.nih.gov/blast/documents/fastacmd.html

2.4 Splitting the query file in small chunks

To run BLAST over cluster we need to split the input fasta file into smaller chunk that will be send over to the cluster for parallel BLAST. We will split the file using a home-grown tool.

```
git clone https://github.com/malaybasu/SeqToolBox.git
```

Now split the fasta file into smaller pieces. The number of piece should approximately equal to the number of nodes that you have in you cluster.

```
echo $(( `bzcat all.fas.bz2| grep \> |wc -l` / 128 ))
perl SeqToolBox/bin/splitfasta.pl -s 86 <(bzcat all.fas.gz)
```

2.5 Submit jobs to cluster

```
for i in `ls *.fas`
do
qsub -cwd -b y -V -j y -o $i.log -l h_rt=1:00:00,vf=5G \
"blastall -e 0.001 -F T -p blastp -i $i -m 9 -d /scratch/user/malay/all \
>$i.bla && bzip2 $i.bla"
done

# Clean the fas files and log files
rm -rf *.log *.fas

# Create a big file containing all the results
cd ..
bzcat test/*.bz2 | bzip2 -c >all.bla.bz2
```

One importatnt option is `-F` that switches the compositional filtering on. I suggest that you keep this option on.

3 HMMER

`HMMER` is the only known software for HMM use in bioinformatics. You can download HMMER from

http://hmmer.janelia.org/software

There are quite a few software that are bundled with HMMER distribution. But the 4 most common ones are:

1. `hmmsearch` - Searches HMMs against protein sequences
2. `hmmscan` - Searches protein sequences against HMM library
3. `hmmbuild` - Builds HMM from a multiple alignment
4. `hmmpress` - Convert a flat file HMM to binary format that can be used with the software

3.1 Exercise

We will search the SwissProt data with a profile of P53 gene.

We will first get a bunch of orthologs of P53 from the Homologene database.

```
wget --quiet ftp://ftp.ncbi.nih.gov/pub/HomoloGene/current/homologene.data
```

Now we will grep the file to find p53:

```
cat homologene.data | grep -i tp53 | head -n 10
```

```
## 460  9606    7157    TP53    120407068   NP_000537.3
## 460  9598    455214  TP53    332847216   XP_001172077.2
## 460  9544    716170  TP53    114051852   NP_001040616.1
## 460  9615    403869  TP53    50978974    NP_001003210.1
## 460  9913    281542  TP53    28849929    NP_776626.1
## 460  10116   24842   Tp53    189083686   NP_112251.2
## 460  7955    30590   tp53    425876787   NP_001258749.1
## 460  8364    431679  tp53    50054422    NP_001001903.1
## 3959 9606    7159    TP53BP2 112799849   NP_001026855.2
## 3959 9598    457766  TP53BP2 332812020   XP_003308815.1
```

Looks like cluster 460 contains T53. We will use a bit of perl code to get the accession.

```
cat homologene.data | perl -ane 'print $F[5],"\n" if ($F[0] == 460)'>p53_homologene_ids.txt
```

We will now use NCBI eutils to extract those sesquence. Let's create a fasta file with this ids:

```
for i in `cat p53_homologene_ids.txt`
do
wget -q -O - "http://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?\
db=protein&id=$i&rettype=fasta&retmode=text"
done >p53.fas
```

We will first use `muscle` to align those sequence and create the HMM, then search SC genome with it.

```
muscle -in p53.fas -out p53.aln
hmmbuild --informat afa p53.hmm p53.aln
hmmsearch -o hits.txt uniprot_sprot.fasta
```