**Scripts\_Final\_assignments\_pcfb**

1 *Regular expression to simplified the header nt fasta files:*

>ref|NZ\_CP004118|gi|759565913|pseudocap|281460 [Burkholderia thailandensis E444 chromosome 2, complete sequence.]

>NZ\_CP004118\_B\_thailandensis\_E444

(>)\w+\|(\w+\_\w+)\|\w+\|\d+\|\w+\|\d+ .(\w)\w+ (\w+) (\w+) .+

\1\2\_\3\_\4\_\5

>ref|NZ\_CP007782|gi|749302224|pseudocap|231576 [Burkholderia cenocepacia strain DDS 22E-1 chromosome 3, complete sequence.]

>NZ\_CP007782\_B\_cenocepacia\_DDS\_22E-1

(>)\w+\|(\w+\_\w+)\|\w+\|\d+\|\w+\|\d+ .(\w)\w+ (\w+) \w+ (\w+) (\w+-\d) .+

\1\2\_\3\_\4\_\5\_\6

>gi|83718394|ref|NC\_007651|pseudocap|178 [Burkholderia thailandensis E264 chromosome I, complete sequence.]

>NC\_007651\_B\_thailandensis\_E264

(>)\w+\|\w+\|\w+\|(\w+\_\d+)\|\w+\|\d+ .(\w)\w+ (\w+) (\w+) .+

\1\2\_\3\_\4\_\5

>gi|170734517|ref|NC\_010512|pseudocap|190 [Burkholderia cenocepacia MC0-3 chromosome 3, complete sequence.]

>NC\_010512\_B\_cenocepacia\_MC0-3

(>)\w+\|\d+\|\w+\|(\w+\_\d+)\|\w+\|\d+ .(\w)\w+ (\w+) (\w+.\w+) .+

\1\2\_\3\_\4\_\5

>gi|Bio::PrimarySeq=HASH(0x69052d0)|pseudocap|283164 [Burkholderia cenocepacia H111 chromosome 3, complete genome]

>Bio::PrimarySeq=HASH(0x69052d0)\_B\_cenocepacia\_H111

(>)\w+\|(\w+::\w+=\w+.+)\|\w+\|\d+ .(\w)\w+ (\w+) (\w+) .+

\1\2\_\3\_\4\_\5

2 *Regular expression to simplified aa sequence:*

>Bcenmc03\_1977 gi|170733313|ref|YP\_001765260|ref|YP\_001765260.1| 2-hydroxy-3-oxopropionate reductase [Burkholderia cenocepacia MC0-3 chromosome 1, complete sequence.]

>Bcenmc03\_1978 gi|170733314|ref|YP\_001765261|ref|YP\_001765261.1| hypothetical protein [Burkholderia cenocepacia MC0-3 chromosome 1, complete sequence.]

>Bcen\_2989 ref|YP\_622858.1|ref|YP\_622858|gi|107025347| NADH:flavin oxidoreductase [Burkholderia cenocepacia AU 1054 chromosome 2, complete sequence.]

(>)\w+\w+ .+\| (.+\w+.+) \[(\w)\w+ (\w+) (\w+.\w+) .+

\1\3.\4\_\5\_\2

>I35\_0072 | virulence associated protein C [Burkholderia cenocepacia H111 chromosome 1 complete genome]

(>)\w+\w+ \| ([A-Za-z].+) \[(\w)\w+ (\w+) (\w+) .+

\1\3.\4\_\5\_\2

>I35\_0154 | 5,10-methylenetetrahydrofolate reductase [Burkholderia cenocepacia H111 chromosome 1 complete genome]

(>)\w+\w+ \| ([A-Za-z1-9].+) \[(\w)\w+ (\w+) (\w+) .+

\1\3.\4\_\2

e.g., *nucleotide sequence simplified*:

>NC\_008061\_B\_cenocepacia\_AU

>NC\_008060\_B\_cenocepacia\_AU

*amino acid sequence simplified:*

>Bcen\_299\_alpha/beta hydrolase

>Bcen\_299\_limonene-1,2-epoxide hydrolase

Explanation:

The regular expression above is pretty straightforward, it is just the amount of data that is huge and so diverse that made the search/replace function a bit tedious job. The other thing, the huge data was made me overwhelmed, which made me use the first possible way that crossed my mind to get this job done, using regular expression rather than maybe using python scripts that might make my job a lot easier. So, the wildcard uses here were \w which matches the character of letter [a-zA-Z0-9], \d a digit [0-9], \s a white space and quantifier + (\w+ and \d+), matches an entire letters and digits before the next non-word characters (such as space, punctuation and the-end-of-the-line). (), uses for the characters we want to capture. Then \1\2\_\3\_\4\_\5 as a replacement term.

3 *Make database for blastall-vs-all analysis:*

$ module load BLAST/2.2.30-goolfc-2.7.11-Python-2.7.3

$ makeblastdb -in all\_proteins.fasta -dbtype prot

Building a new DB, current time: 01/27/2016 10:36:06

New DB name: all\_proteins.fasta

New DB title: all\_proteins.fasta

Sequence type: Protein

Keep Linkouts: T

Keep MBits: T

Maximum file size: 1000000000B

Explanation:

All-vs-all Blast is a common method used to search for protein orthologs in a protein database. This method is known to find new protein families, and to give about homology clustering relationship (Prince et al 2008). Prior analysis, the protein database should be build based on the overall amino acid input data, for thus will be use to do blast. To do so, first thing we should do is to load the module (type of blast program in the peregrine environment), next is to make the database in all\_proteins.fasta file in format which blast module will recognize when do the analysis (-dbtype prot).

4 *Make scripts for job submission (BLAST):*

#!/bin/bash

#SBATCH --job-name=blast-all-vs-all

#SBATCH --nodes=1

#SBATCH --ntasks-per-node=12

#SBATCH --time=2-23:00:00

#SBATCH --partition=himem

#SBATCH --mem=240GB

cd /home/p272779/scripts

module load BLAST/2.2.30-goolfc-2.7.11-Python-2.7.3

blastp -db all\_proteins.fasta -query all\_proteins.fasta -outfmt 6 -out all-vs-all.tsv -num\_threads 12 -evalue 1e-5 -num\_alignments 1

Explanation:

The job needs to run with such parameters in order to make the job run faster. It might be overestimated parameters, but that in my opinion indeed made the job run faster.

The module/program used was BLAST/2.2.30-goolfc-2.7.11-Python-2.7.3, which already available in peregrine cluster. The input file is all\_proteins.fasta, which will create all-vs-all.tsv file as an output file.

The scripts parameter are that the blast should have an E-value 1-e5, and BLAST maximum target sequence number 1, as we wanted only the best blast hits possible.

5 *Transfer file from your computer to peregrine.hpc.rug.nl:*

scp -r ./gbk/ p272779@peregrine.hpc.rug.nl:/home/p272779/

p272779@peregrine.hpc.rug.nl's password:

e.g.,

Burkholderia\_ambifaria\_AMMD\_128.gbk 100% 17MB 8.6MB/s 00:02

Burkholderia\_ambifaria\_MC40-6\_136.gbk 100% 17MB 17.4MB/s 00:01

Burkholderia\_cenocepacia\_AU\_1054\_129.gbk 100% 17MB 16.7MB/s 00:01

Burkholderia\_cenocepacia\_DDS\_22E-1\_2504.gbk 100% 15MB 15.5MB/s 00:01

Burkholderia\_cenocepacia\_DWS\_37E-2\_2505.gbk 100% 13MB 12.7MB/s 00:01

Explanation:

Once the job done, and all-vs-all.tsv file as the output file been created, the next step is to transfer this output file to the local computer, where the next analysis, which is the clustering and identification of core genome will be done. $ scp stands for secure copy, which usually use to copy file/directory between local computer to remote system (in this case peregrine) or between two remote systems.

6 *Job submission:*

sbatch all-vs-all.sh

Submitted batch job 3636885

7 *Check the submitted job:*

$ squeue -u p272779

JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)

$ ls

SLURM\_pcfb\_ex1.sh all-vs-all.tsv all\_proteins.fasta.phr all\_proteins.fasta.psq all-vs-all.sh all\_proteins.fasta all\_proteins.fasta.pin

Note. Underline is database; Yellow is file.

Explanation:

The job runs for approximately 2 hours usuing such parameters. And all-vs-all.tsv file was created as the output file.

Output:

B.cenocepacia\_AU B.cenocepacia\_HI2424 100.00 436 0 0 1 436 1 436 0.0 895

B.cenocepacia\_AU B.cenocepacia\_HI2424 100.00 306 0 0 1 306 1 306 0.0 613

B.cenocepacia\_AU B.cenocepacia\_MC0-3\_short-chain 100.00 245 0 0 1 245 1 245 1e-173 483

B.cenocepacia\_AU B.cenocepacia\_MC0-3\_NADH:flavin 100.00 360 0 0 1 360 1 360 0.0 724

B.cenocepacia\_AU B.cenocepacia\_MC0-3\_alpha/beta 100.00 278 0 0 1 278 1 278 0.0 572

B.cenocepacia\_AU B.cenocepacia\_MC0-3\_limonene-12-epoxide 100.00 129 0 0 1 129 1 129 8e-90 261

B.cenocepacia\_AU B.cenocepacia\_HI2424 100.00 399 0 0 1 399 1 399 0.0 820

B.cenocepacia\_AU B.cenocepacia\_HI2424 100.00 443 0 0 1 443 1 443 0.0 879

B.cenocepacia\_AU B.cenocepacia\_MC0-3\_LysR 100.00 306 0 0 1 306 1 306 0.0 615

B.cenocepacia\_AU B.cenocepacia\_MC0-3\_extracellular 100.00 257 0 0 1 257 1 257 0.0 519

B.cenocepacia\_AU B.cenocepacia\_MC0-3\_succinylglutamate 100.00 371 0 0 1 371 1 371 0.0 744

B.cenocepacia\_AU B.cenocepacia\_MC0-3\_IclR 100.00 275 0 0 1 275 1 275 0.0 560

B.cenocepacia\_AU B.cenocepacia\_MC0-3\_beta-ketoadipyl 100.00 400 0 0 1 400 1 400 0.0 805

B.cenocepacia\_AU B.cenocepacia\_HI2424 100.00 198 0 0 1 198 1 198 7e-142 399

B.cenocepacia\_AU B.cenocepacia\_HI2424 100.00 186 0 0 1 186 1 186 5e-133 375

Explanation:

Above is the output file from blast all-vs-all analysis. What is important are the first lines is the query sequence and the second is the best match sequence from data best. Third column is the percentage identity and column before end is the e-value. The first two columns and e-value will then be use for gene family clustering using MCL program.

8 *Transfer data from peregrine to computer:*

In home (computer) directory:

$ scp p272779@peregrine.hpc.rug.nl:/home/p272779/scripts/all-vs-all.tsv ~/Documents/pcfb/Project

p272779@peregrine.hpc.rug.nl's password:

all-vs-all.tsv 100% 7641KB 1.1MB/s 00:07

$ cd ~/Documents/pcfb/Project/

$ ls

Final\_report.docx all-vs-all.tsv faa\_copy fna\_copy

Report.docx faa fna gbk

Explanaton:

As aforementioned, to do the next analysis that is the gene family clustering, the output file created using blast all-vs-all analysis should be transfer to local computer, which the MCL program is located.

9 *Identify the core genome:*

Core is 80% similarity in aa sequence

$ cd ~/Scripts

$ nano

#!/usr/bin/env python

Filename= "all-vs-all.tsv"

File= open(Filename, 'r')

Outfilename= "Outallvsall.tsv"

Outfile=open(Outfilename, 'w')

for line in File:

value =line.split('\t')

val = int(float(value[2]))

if(value >=80):

print line

Outfile.write(line)

File.close()

Outfile.close()

Explanation:

Core genome is the sets of genes that present in all the species strain. I set the criteria for identifying the core gene which is the value of percentage identity of and more than 80% similarity of amino acid sequence.

10 *Clustering:*

(<http://micans.org/mcl/index.html?sec_news>)

Installation:

cd ~/Downloads/mcl-14-137/

./configure --prefix=$HOME/local

make install

$ cut -f 1,2,11 all-vs-all.tsv > all-vs-all.abc

$ mcxload -abc all-vs-all.abc --stream-mirror --stream-neg-log10 -stream-tf 'ceil(200)' -o all-vs-all.mci -write-tab all-vs-all.tab

$ mcl all-vs-all.mci -I 1.4 -use-tab all-vs-all.tab

$ mcl all-vs-all.mci -I 2 -use-tab all-vs-all.tab

$ mcl all-vs-all.mci -I 4 -use-tab all-vs-all.tab

$ mcl all-vs-all.mci -I 6 -use-tab all-vs-all.tab