**GeneratingRefSeq: Generation of a near complete genome sequence database**

***User Manual***

***Introduction***

This pipeline allows for a text file of FASTA sequences to be generated, which encompasses the virus species or genus diversity, which can be used for virus identification in reference-based assembly approaches. The first step is to download all sequences associated with a taxonomic ID, which are then filtered by various parameters with the most important being sequence length. This is then used to align and draw a phylogenetic tree and cluster sequences, which when combined allows for a scatterplot to be generated of root-to-tip divergence vs time in addition to creating a file of selected sequences (based on a specified year).

***Installation***

Prior to running the pipeline or as individual scripts, the following programs need to be installed in either your working directory or added to your path.

* Python 2.7.11 or above (<https://wiki.python.org/moin/BeginnersGuide/Download>).

Although not a prerequisite as python packages can be downloaded from source, the python software management tool pip (already available on 2 >=2.7.9 or Python 3 >=3.4) (<https://pip.pypa.io/en/stable/installing/>) allows for easy installing using:

pip install package Name

* Biopython (<http://biopython.org/wiki/Download>).
* MAFFT (<http://mafft.cbrc.jp/alignment/software/installation_without_root.html>)
* CDHIT-est ([http://weizhongli-lab.org/lab-wiki/doku.php?id=cd-hit-user-guide#installation](http://weizhongli-lab.org/lab-wiki/doku.php?id=cd-hit-user-guide" \l "installation)) under ‘Installation’.
* FastTree (<http://meta.microbesonline.org/fasttree/#Install>).
* Ete3 Toolkit (<http://etetoolkit.org/download/>).
* Fasta-formatter(<http://hannonlab.cshl.edu/fastx_toolkit/download.html>) where the appropriate version should be downloaded, uncompressed and installed for Unix as shown in the instructions (<http://hannonlab.cshl.edu/fastx_toolkit/install_ubuntu.txt>).

The individual modules, which are required for the pipeline and individual scripts to run(however the python interpreter looks for these independently) and found after the import statement or specific attributes (from Module import attribute) at the top of each script are:

subprocess, csv, argparse, Bio(attribute: SeqIO), re, ete3(attribute:Tree), plotly.offline(attribute: plot), plotly.graph\_objs(attributes; \*(all)), plotly, collections (attribute: OrderedDict).

***Running the pipeline***

The pipeline wrapper and scripts required, ((need to be created using your desired text editor e.g. vim)) for the pipeline to run, which need to be in your working directory or added to your path are:

GeneratingRefSeq.py (pipeline)

NCompleteGenomes.py

Combo.py

FilteredSeq.py

GetRefseq.py

Plotly.py

A text file format (“Virus.txt”) has also been supplied which needs to be filled in with the appropriate taxonomic ID, near complete genome threshold (defined as 90% of coding region of a RefSeq sequence) and a year cut off after which sequences are wanted. Example in the supplied file is Yellow Fever Virus. CAUTION: Ensure that only a tab is between each variable.

Once this has been entered, type on the command line:

python GeneratingRefSeq.py

Once running messages will appear showing the progress of the script and if any errors have occurred.

Typing python GeneratingRefSeq.py -h or --help will give a description of the pipeline and what is required as input.

***Running individual scripts***

To run individual scripts within your working directory type on the command line the relevant one with arguments(shown in italics and blue below) filled out appropriately (if not entered then a help command will become visible):

E-utilities

esearch -db nucleotide -query txid*taxonomidID*[Organism]| efetch -format gb > *Output file name*

esearch -db nucleotide -query txid*taxonomicID*[Organism]| efetch -format fasta > *Output file name*

NearCompletGenomes.py

python NCompleteGenomes.py -i *Genbank file* -*o Output file name* -l *near complete genome threshold*

CD-HIT

cd-hit-v4.6.5-2016-0304/cd-hit-est -i *Filtered FASTA file* -o *Output file name* -c 0.9 -n 8

MAFFT

mafft --auto *Filtered FASTA file* > Output file name

FastTree

FastTree -nt *MAFFT file* > *Output file name*

Combo.py

python Combo.py -i *CDHIT .clstr file* -i2 *FastTree output file* -o *Output file name*

FilteredSeq.py

python FilteredSeq.py -i *Output file from previous script* -o *Output file name* -y *user chosen year cut off*

GetRefseq.py

python GetRefseq.py -i *Output file from previous script* -i2 *FASTA file from E-Utilities* -o *Output file name*

Plotly.py

python Plotly.py -i *Output from Combo.py script*

***Outputs***

The outputs from the pipeline are placed in a folder titled with the associated taxonomic ID for example for Yellow fever virus will be ‘11089’ which will contain:

11089.gb (Genbank files from NCBI)

11089.fasta(FASTA files from NCBI)

11089.fa (Filtered FASTA file based on near complete genome length)

11089.tsv (tab delimited file from NCompleteGenomes.py script)

cdhit11089 (representative sequences from each cluster)

cdhit11089.clstr (all accession numbers for each cluster and similarity to representative)

mafft11089.fa (alignment)

Combo11089.tsv (tab delimited file from Combo.py script)

Filtered11089.tsv (tab delimited file from FilteredSeq.py script)

temp-plot.html (interactive sccatterplot)

refseq11089.txt (selected sequences in FASTA format)

lin11089.fasta (linearized format of original FASTA file to allow grep command to work).

logfile.txt (Giving the number of sequences downloaded, number of sequences aligned and clusters formed for a given taxonomic ID).