Metagenomics tutorial

Dr Cristina Venturini

The purpose of this document is to provide a brief introduction to metagenomics analysis.

You are an epidemiologist on a field trip in the Democratic Republic of the Congo. A cluster of cases of patients with haemorrhagic symptoms and fever has occurred in a remote village and 2 nurses have developed similar symptoms and been admitted to the local hospital. Samples from the nurses have been sent to the East Africa reference laboratory at the Uganda Virus Research Institute and one has been deep sequenced on the Illumina MiSeq platform. You need to analyse the data to determine the cause of the outbreak and determine what treatment options may be available.

Note that metagenomic analysis can be very time consuming and typically requires more computational resources than are available on the VMs used on this course. Therefore, some steps have been performed for you and we will be working with output files from some analyses.

Befor you start...

You are welcome to copy and paste the commands, but please take your time to read carefully and understand what we are doing. Spend some time reading the main page of the softwares used (for example Centrifuge).

Backslash in Linux

```
\
```

You will see I use the backslash a lot in the tutorial. It allows a command to span multiple lines to make it easier to read and type. Some info here: https://www.cyberciti.biz/faq/howto-ask-bash-that-line-command-script-continues-next-line/

First step: Assembly

A bioinformatician has already performed metagenomic de novo assembly on the sequencing reads using MetaSPAdes (http://cab.spbu.ru/software/spades/) – this took

several hours. The command used is below (do not run this command – it will not complete during the practical).

```
#DO NOT RUN THIS
~/Programs/SPAdes-3.14.0-Linux/bin/spades.py --meta -k 21,33,55,63 \
-t 2 -m 8 -1 \
~/Cristina/Metagenomics/Raw_reads/SRR533978_1.fastq.gz -2 \
~/Cristina/Metagenomics/Raw_reads/SRR533978_1.fastq.gz \
-o ~/Cristina/Metagenomics/Assemblies
```

NB. if you only have a file, you can use flag --12 instead of -1 and -2 to specify the files.

Create a directory for the tutorial

```
mkdir Metagenomics_Training

cd Metagenomics_Training

cp ~/Cristina/Metagenomics/Assemblies/metaspades-raw.contigs.fasta ./
```

You can look at the contig sequences:

```
less metaspades-raw.contigs.fasta
```

```
>NODE_1_length_5610_cov_41.937534
GTTTCCCACTGGAGGATACGCGGCGACGGGAAAATTGCATTTCAATTTGGAAATTCGATC
GTTCCACTAGATATTACAAATCATCGAGTAAGAGAATGGCAAATCAAGCTATTCTCCGCG
TCGTGGCAACCAGATATCGGGTTGGACAATTTTCAGTTAGGCGATCGAGAAAGAGTCTGC
GTAGAAAAAATCCAGAACAACGACTACATACAAAACGACAAAGAAACCGATGATCGAACG
AAATTCAGAAATGCAAAATTGATTTGGCTGGAGGGGGTAGCACAGAGTTTACACAAAATC
GGGAAAACCTTAACAGTCAACTATTCGCTTGATGTCCCGCCTGATTCTCAATCTGCAAAG
CGGCTGTTGAATTCCGTCGATGGCGTGTTGTACGAAGACTATTGGGGGGACCAATTCAAA
AGGGGAATGTTTTCCAGCACAAATCGTATGCTTGATTTACAAAAATTCTTCTCCGACGCA
ACGAAAAAAAACAAACGACTTTATTTTATATTTCAACTTCCAGAGGTTAATAACGAAAAC
ATCCAAACCGCAATGGCTGCCTATCTCATCAATTCGAGTTCACAGTCTGCAGTCTACATA
TCGGCAGTTCAACGTTATGGAATCGTTCCCGGCTTCTATGGCTACGACAAAACGATCGGC
GACGCATGCGGGCCAGCGGTAATCGGTCATGAAACTATCGTCCGGCAATTCGCTGGTGCG
```

Or you can look at the contig headers:

```
grep '>' metaspades-raw.contigs.fasta | less
```

```
>NODE_35_length_1880_cov_68.171507
>NODE_36_length_1862_cov_28.728832
>NODE_37_length_1816_cov_41.756956
>NODE_38_length_1814_cov_20.002274
>NODE_39_length_1774_cov_25.461896
>NODE_40_length_1768_cov_36.720374
>NODE_41_length_1755_cov_40.171176
>NODE_42_length_1743_cov_54.174763
>NODE_43_length_1737_cov_24.470868
>NODE_44_length_1723_cov_19.685851
>NODE_45_length_1718_cov_34.026458
```

Each header contains the name of the contig ('NODE*X'), the sequence length ('length*X') and a measure of coverage ('cov_X'; note that this is an output of SPAdes and does not represent the true coverage, but the number of k-mer hits from the final iteration of k). Try

scrolling through the list to look at the size and coverage of the contigs.

How many contigs have been assembled? You can count the number of contigs in the assembly:

```
grep -c '>' metaspades-raw.contigs.fasta
```

The contigs in this file have been ordered by size, with the largest at the top. Scroll down to the bottom – most contigs are very small (less than the length of a sequencing read). These are unlikely to be useful. We will use a short script to remove small contigs, allowing us to narrow our search by limiting contigs to those > 500bp in length:

```
~/Cristina/Metagenomics/scripts/remove_small_contigs.pl 500 \
./metaspades-raw.contigs.fasta > \
metaspades-raw.contigs.filtered.fasta
```

How many contigs do we have left? Try modifying the "grep -c" command we used earlier.

Second step: Contig Classification - Viral RefSeq

We will now try and identify the species present in our metagenomics assembly. We will be using Centrifuge (https://ccb.jhu.edu/software/centrifuge/) for this tutorial - it is fast and computationally lightweight. There are many other softwares (i.e. Kraken, Metamix).

We will first use Centrifuge to compare our filtered contigs against a database of all published Viral reference sequences (Viral RefSeq):

```
centrifuge -p2 -f \
-x ~/Cristina/Metagenomics/Centrifuge_Db/Centrifuge-viral_db \
./metaspades-raw.contigs.filtered.fasta \
-S ./contigs.filtered.viral-refseq.centrifuge
```

```
training@bioinformatics:~/Training/Cristina/Metagenomics_Training$ centrifuge -p2 -f \
> -x ~/Training/Cristina/Metagenomics/Centrifuge_Db/Centrifuge-viral_db \
> ./metaspades-raw.contigs.filtered.fasta \
> -S ./contigs.filtered.viral-refseq.centrifuge
report file centrifuge_report.tsv
Number of iterations in EM algorithm: 0
Probability diff. (P - P_prev) in the last iteration: 0
Calculating abundance: 00:00:00
```

```
mv centrifuge_report.tsv \
contigs.filtered.viral-refseq.centrifuge-report.tsv
```

Take a look at the directory:

```
ls -lhrt
```

Centrifuge has produced two files:

- ***contigs.filtered.viral-refseq.centrifug*** : a list of each classified contig and its classification
- ***contigs.filtered.viral-refseq.centrifuge-report.tsv***: a summary of the different species assignments found by Centrifuge

Take a look at each file (we can use the following command to make the text file slightly easier to read):

```
cat contigs.filtered.viral-refseq.centrifuge | \
perl -pe 's/ /_/g' | head

cat contigs.filtered.viral-refseq.centrifuge-report.tsv | \
perl -pe 's/ /_/g'| less
```

```
name    taxID   taxRank genomeSize      numReads        numUniqueReads  abundance
Shamonda_orthobunyavirus        159150  species 12104   1       1       0.0
Choristoneura_occidentalis_granulovirus 364745  species 104710  4       4       0.0
Burkholderia_phage_KS14 910475  species 32317   1       1       0.0
Salmonella_phage_SJ46   1815968 species 103445  1       1       0.0
Tokyovirus_A1   1826170 species 372707  1       1       0.0
Escherichia_virus_M13   1977402 species 6407    3       3       0.0
```

Each contig has been assigned a closest matching sequence ID (seqID) from the reference database, along with a taxonomic identifier (taxID). We also have information about the quality of the sequence hit and the length of the match. The contigs.filtered.viral-refseq.centrifuge-report.tsv file summarises this information and expands the 'taxID' to give the scientific name for each species (name).

Take a look through the species identifications. Look up any species you are unfamiliar with online. Do any of these seem like a potential causative agent for these patients? What might explain some of the hits detected in this scenario (e.g. Salmonella_phage)?

How many reads have been successfully classified? How many contigs did you submit for analysis (you worked this out earlier)? What does this tell you about the Viral RefSeq database we used and the contigs we have sequenced?

Third step: contig classification - whole Nt database

We will now use a much larger database (the total 'nt' or non-redundant nucleotide database) downloaded from NCBI. This database is very large and should contain all known sequences – not just those formally classified as Reference Sequences. However, not all sequences will have appropriate taxonomic assignments. Because this database is so large,

it will not be possible to either build the database, or to run it on the local VMs (memory requirements to build this database were 440GB and to run the analysis with the Nt database it took 128GB RAM).

Copy the Centrifuge output files into your current working directory:

```
cp ~/Cristina/Metagenomics/Full_centrifuge_screens/\
metaspades-raw.contigs.filtered.centrifuge_nt_db* ./
```

We no longer have a report file (due to most sequences failing taxonomic assignment). However, we can still take a look at the read assignment outputs as previously:

```
cat metaspades-raw.contigs.filtered.centrifuge_nt_db \
| perl -pe 's/ /_/g'| \
column -t | less
```

How many contigs have been matched now?

```
grep -c 'NODE' metaspades-raw.contigs.filtered.centrifuge_nt_db
```

Contigs with assigned matches have an entry in the 'seqID' column. Note that the 'taxID' column is now empty. We can investigate the hits in a number of ways:

1. We can explore the raw contig hits – a good starting point would be to look at contigs with the greatest length and coverage
2. We can look at species for which we have multiple contigs. We can also work out species hits that occur for multiple contigs using the following command:

```
cat metaspades-raw.contigs.filtered.centrifuge_nt_db | \
perl -lane 'print "$F[1]"'| \
sort | uniq -c | sort -rn | less
```

Look through the data and try to identify hits that have both long contigs and multiple hits. Use Google or NCBI nucleotide to work out what species the seqID comes from.

Retrieval of contigs of interest from full assembly

Having identified potential viruses of interest, we can retrieve all contigs that have been matched to that reference. First, we create a list of contigs by extracting the first column from matching lines:

```
grep 'JX297815.1' metaspades-raw.contigs.filtered.centrifuge_nt_db | \
perl -lane 'print "$F[0]"' > contigs-of-interest.list
```

Where 'JX297815.1' is the seqID we wish to extract.

Then, we use seqtk (https://github.com/lh3/seqtk) to extract the contigs in our list from the full

file:

```
~/Programs/seqtk/seqtk subseq \
metaspades-raw.contigs.filtered.fasta contigs-of-interest.list >\
contigs-of-interest.fasta
```

Take a look at the file we have created:

```
less contigs-of-interest.fasta

grep '>' contigs-of-interest.fasta
```

Exploring match quality using web-based BLAST.

We can explore the quality of the matches for our interesting contigs using BLAST. We will do this using an online version of BLAST (we can use the version either at EBI or at NCBI).

1. In your web browser, navigate to https://blast.ncbi.nlm.nih.gov/Blast.cgi
2. Select "Nucleotide BLAST"
3. Cut and paste a sequence from your ‘contigs-of-interest.fasta’ file into the sequence input area on the page
4. Select BLAST - Submit (and wait)
5. Explore the BLAST web page output. What does the analysis tell you about your match?

Try putting some other contigs into BLAST (both of your ‘seqID of interest’ and for other contigs)

Retrieve appropriate genome sequence and map reads back to confirm presence

You should have a good hit for at least one contig with a plausible candidate virus. We will now retrieve the reference sequence for that hit and map the sequencing reads back to confirm that the candidate virus is present in our sample.

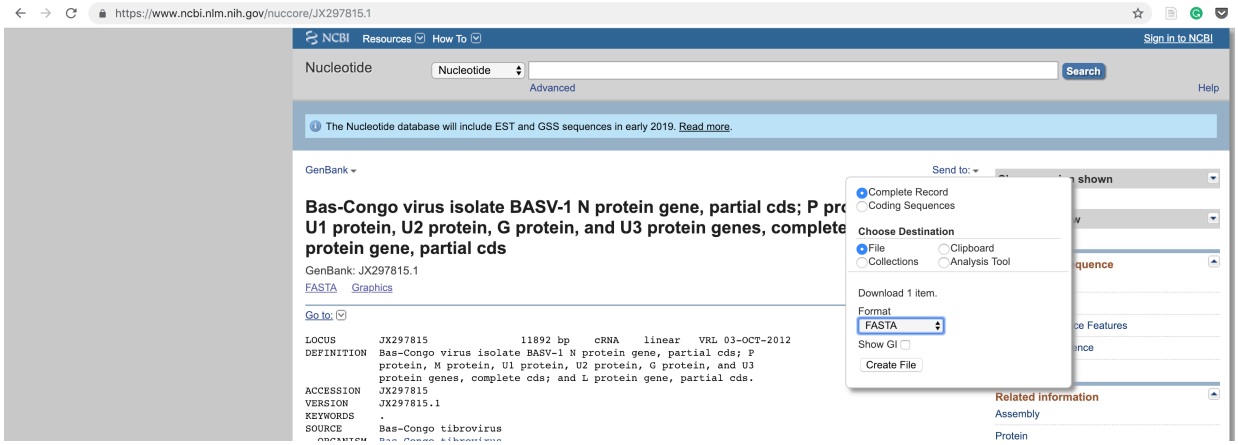
I pre-prepared the reference genome:

```
cp ~/Cristina/Metagenomics/ref_genomes/myreference.fasta ./
```

If you feel bold and you want to get the reference genome yourself:

1. Navigate your browser to https://www.ncbi.nlm.nih.gov/nucleotide?cmd=search and input the reference sequence of interest into the search box (hint:JX297815.1)
2. Click on "Send to", select "File", select format "FASTA", click "Create File"
3. A file should download to the desktop on your VM.

4. You will need to copy this file into your current directory, which should be: /home/training/Metagenomics_Training/



Now we can map our raw sequencing reads back to the reference genome:

```
bwa index myreference.fasta

bwa mem -t 2 -M -T 15 -W 25 ./myreference.fasta \
~/Cristina/Metagenomics/Raw_reads/SRR533978_1.fastq.gz \
~/Cristina/Metagenomics/Raw_reads/SRR533978_2.fastq.gz| \
samtools view -@ 2 -b -f 2 - | samtools sort -@ 2 - \
> myreference.map.bam
```

We can check how well the reads have mapped (note that we could have marked duplicates if we wanted earlier):

```
   samtools flagstat myreference.map.bam
```

Addendum and notes:

It is important to note that the workflow we have followed is an abbreviated version and should not necessarily be taken as a ‘how to’ for detecting underlying pathogens in such cases. A number of steps were omitted for brevity (e.g. read trimming, qc, assembly qc). An alternate approach might have been to perform metagenomic classification of reads (rather than contigs) – this was omitted both due to time constraints, but also due to the increased likelihood of detecting spurious results. This practical assumes that the virus causing disease actually exists in ANY sequence database. In a real situation, it is possible that the ‘reference based classification’ method we used here (i.e. Centrifuge) would not work simply because the causative virus is completely novel. In such a situation, tBLASTx based comparison of contigs might be a better strategy. Alternatively, reference-free binning strategies might be successful, particularly if multiple patients were sequenced, allowing a co-assembly (e.g. MetaBAT2, Concoct).

Conclusions: What is the likely cause of the outbreak?