Metagenomics tutorial

Dr Cristina Venturini

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

We'll be analysing a clinical specimen from a 41-years-old patient with no history of hepatitis, tubercolosis or diabetes. The patient reported fever, chest tightness, cough, pain and weakness and was admitted to hospital 6 days after the onset of the disease. Preliminary aetiological investigations excluded the presence of influenza virus, *Chlamydia pneumoniae* and *Mycoplasma pneumoniae* and this was confirmed by PCR. Other common respiratory pathogens, including human adenoviruses, also tested negative. To investigate the possible aetiological agents associated with this disease, bronchoalveolar lavage fluid (BALF) was collected and deep meta-transcriptomic was performed.

You need to analyse the data to determine the cause of the disease 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/

Data

Our dataset comes from a bronchoalveolar lavage fluid (BALF) clinical specimen. Total RNA was extracted from 200 µl of BALF and a meta-transcriptomic library was constructed for pair-end (150-bp reads) sequencing using an Illumina MiniSeq. For the tutorial, the original fastq data has been downsampled. The original data are available fron the NCBI Sequence Read Archive (SRA) database under BioProject accession number PRJNA603194 (run SRR10971381).

First step: Quality Control

The first step of any analysis should be to check and improve the quality of our data. This step was already performed for you. The original data included 56,565,928 sequence reads and required a lot of computational power/time. Usually we performed two different quality control and filtering before starting any analysis:

- traditional QC for fastq files as seen earlier in this course (ie. remove adapters and low-quality bases)
- dealing with host contamination (filtering out human reads)

There are pipelines available for this step, here I used Metagen-FastQC https://github.com/Finn-Lab/Metagen-FastQC.

This pipeline uses some of the tools you have seen in this course: trim_galore, BWA, samtools and bedtools. Here's an example of how to run Metagen-FastQC where **-t** controls the number of threads; **-f** can be either the forward read file or just a single-end FASTQ file. When using paired-end files, you also use **-r** to point at the reverse file. **-c** is the host genome reference (in our case the human genome hg38.fa).

```
#DO NOT RUN THIS
metagen-fastqc.sh -t 8 \
-f SRR10971381_1.fastq.gz -r SRR10971381_2.fastq.gz \
-c hg38.fa
```

Second step: De Novo Assembly

At this point, we have clean reads that are ready for assembly to contigs. 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
spades.py --meta -k 21,33,55,63 \
-t 2 -m 8 \
-1 ~/Cristina/Metagenomics/Reads/SRR10971381_clean_sub_1.fastq.gz \
-2 ~/Cristina/Metagenomics/Reads/SRR10971381_clean_sub_2.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:

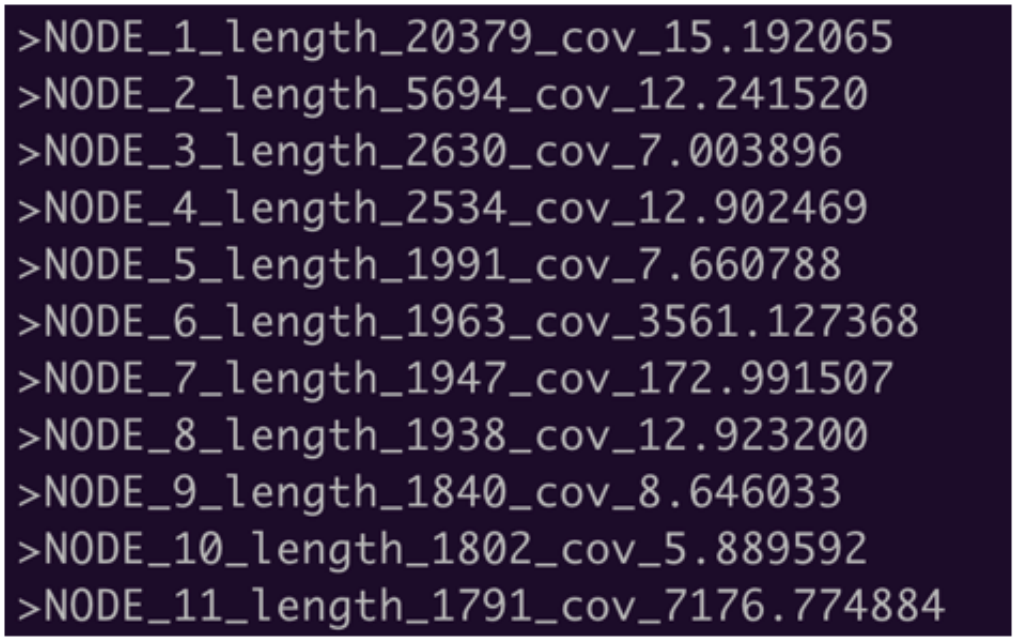
```
mkdir Metagenomics_Training

cd Metagenomics_Training

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

You can look at the contig sequences:

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

Or you can look at the contig headers:

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



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.

Third step: Contig Classification

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
```

```
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.centrifuge*** : 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:

```
cat contigs.filtered.viral-refseq.centrifuge | head
```

```
readID  seqID   taxID   score   2ndBestScore    hitLength       queryLength     numMatches
NODE_2_length_5694_cov_12.241520        NC_004718.3     227859  361     169     34      5694    1
NODE_3_length_2630_cov_7.003896 NC_014470.1     864596  314     0       75      2630    1
NODE_4_length_2534_cov_12.902469        unclassified    0       0       0       0       2534    1
NODE_5_length_1991_cov_7.660788 unclassified    0       0       0       0       1991    1
NODE_6_length_1963_cov_3561.127368      unclassified    0       0       0       0       1963    1
NODE_7_length_1947_cov_172.991507       unclassified    0       0       0       0       1947    1
NODE_8_length_1938_cov_12.923200        unclassified    0       0       0       0       1938    1
```

```
cat contigs.filtered.viral-refseq.centrifuge-report.tsv | head
```

```
name    taxID   taxRank genomeSize      numReads        numUniqueReads  abundance
T4virus 10663   genus   167473  1       0       0.0
SARS_coronavirus        227859  leaf    29751   2       2       0.0
Pseudomonas_virus_F116  280701  species 65195   1       1       0.0
Streptococcus_phage_PH10        644007  species 31276   2       2       0.0
Pseudomonas_phage_PaP1  685892  leaf    91715   1       1       0.0
Cyanophage_S-RIM50      687803  species 174307  4       0       0.0
Bat_coronavirus_BM48-31/BGR/2008        864596  species 29276   2       2       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?

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 contigs.filtered.viral-refseq.centrifuge | \
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 'NC_004718.3\|NC_014470.1' \
contigs.filtered.viral-refseq.centrifuge | \
perl -lane 'print "$F[0]"' > contigs-of-interest.list
```

Where **NC_004718.3** and **NC_014470.1** 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:

```
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)

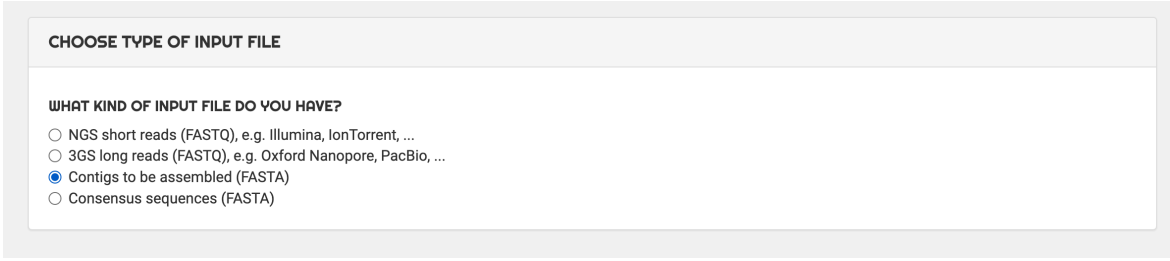
Web-based analysis

It is important to consider that the workflow we used here is just an example of how to perform metagenomics analysis. Alternative approaches do exist, for example we could have performed metagenomic classification of reads (rather than contigs). However, this approach would increase the likelihood of detecting spurious results.

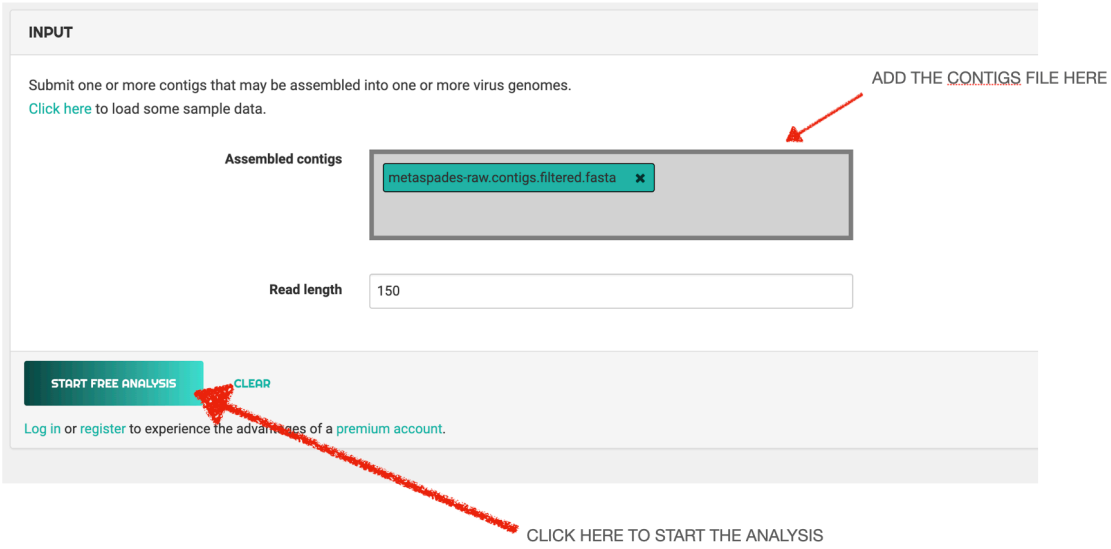
We will try now to use a user-friendly web-based workflow which can be used both for reads and contigs.

For this we use https://www.genomedetective.com/app/typingtool/virus/. The Virus tool in Genome Detective assigns taxonomic names to sequences from eukaryotic viruses and phages. The taxonomic rank which gets assigned is based on availability of reference genomes in RefSeq, and is mostly at species level. At the moment 11140 distinct taxonomic names are assigned based on 14500 reference sequences. Subtyping tools, for the identification of subspecies, are available for 19 viruses. As input we use the contigs file we onbtained after Metaspades and after we filtered for contigs length: **metaspades-raw.contigs.filtered.fasta**.

1. In your web browser, navigate to https://www.genomedetective.com/app/typingtool/virus/
2. What kind of input file do you have? Choose "contigs to be assembled (FASTA)"

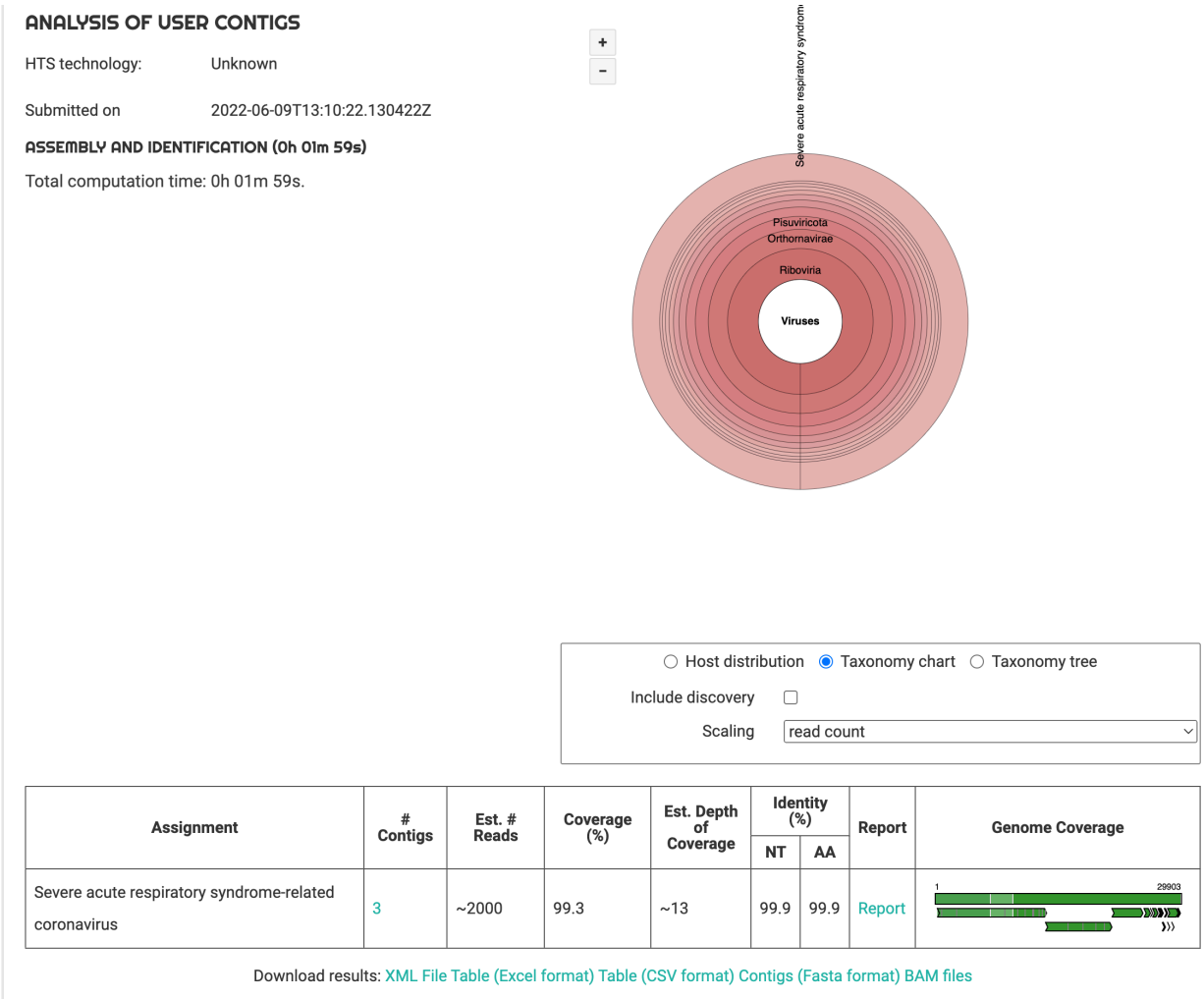


3. INPUT: Upload metaspades-raw.contigs.filtered.fasta in **Assembled contigs**. Put 150 in **Read length**. Click start analysis.



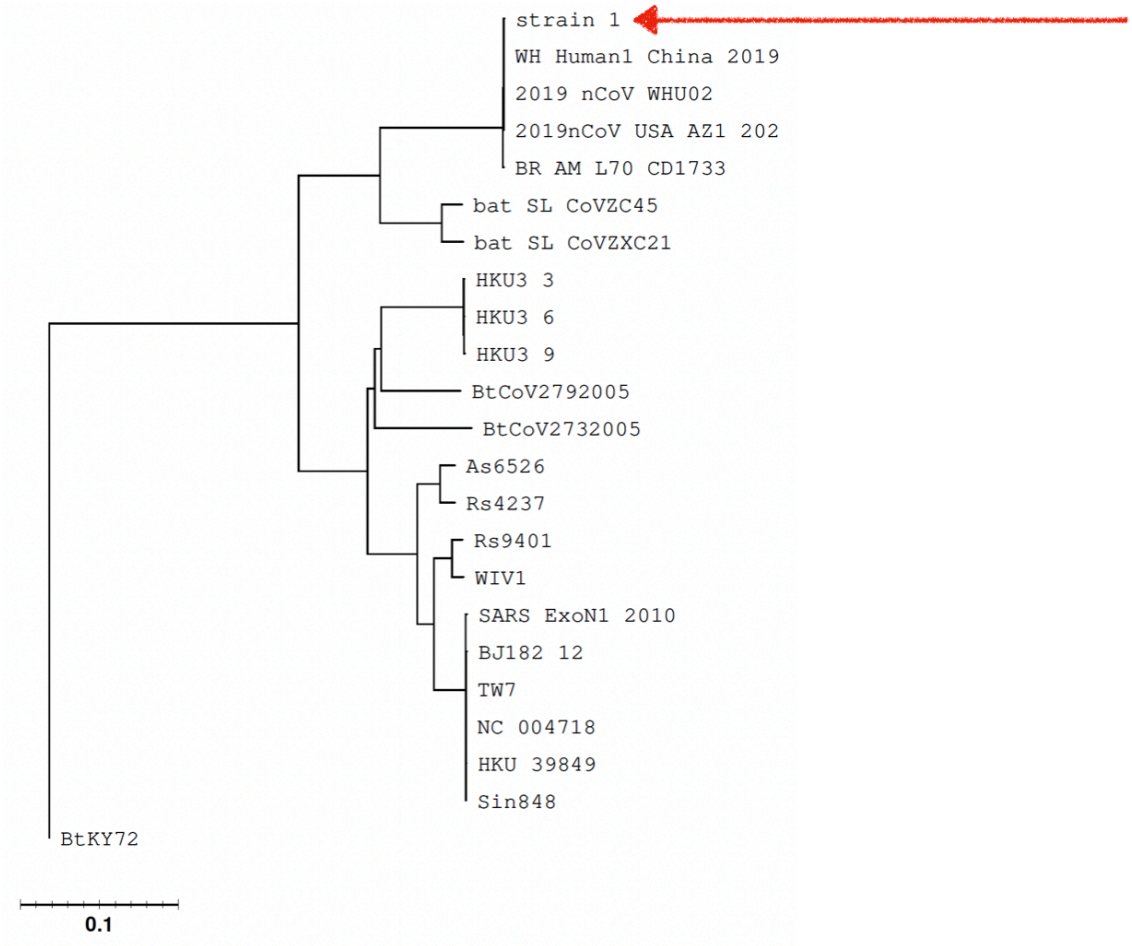
4. Analysis should take only a couple of minutes

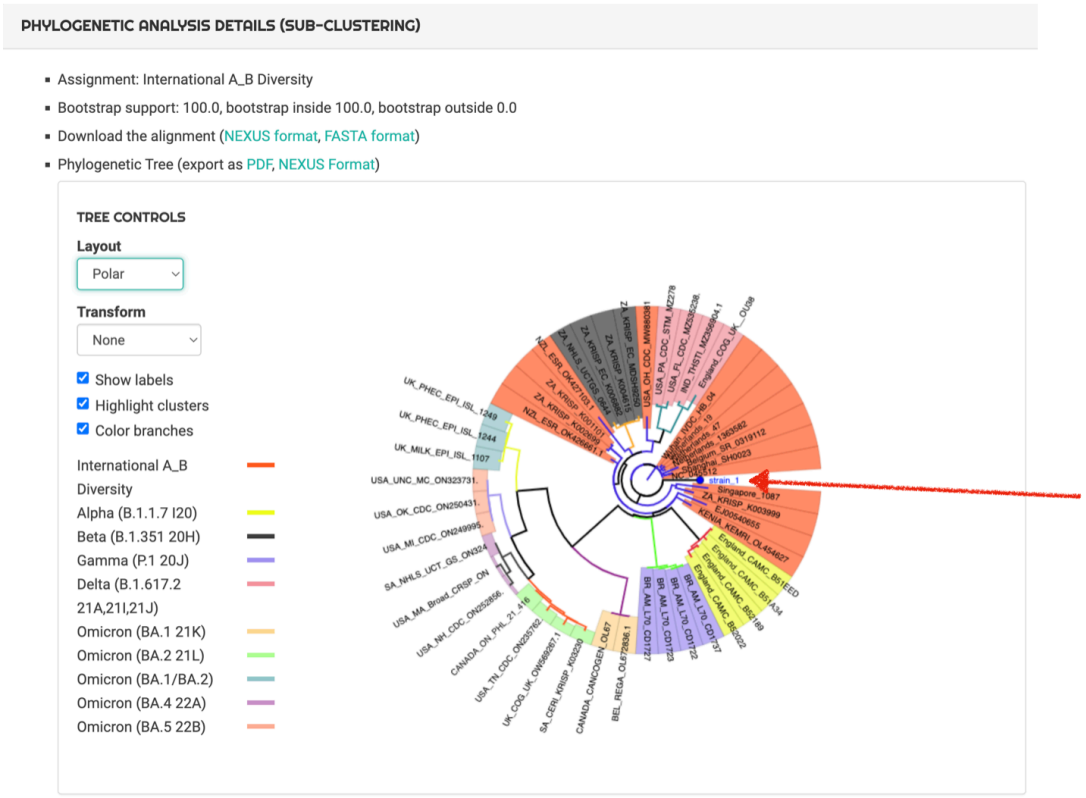
Let's have a look at the results:



The initial page shows a summary of the results. You can see the assignment made, the number of contigs, the coverage %, the estimated depth of coverage, identity and genome coverage.

Click on **Report** and you will get more details about the NGS and alignment. You can continue to explore the Virus tool , for example clicking on "Continue to the cov typing tool" and exploring the genotype and phylogenetic analysis. You will see phylogenetic analysis in details in another module, but for now you can have a look the phylogenetic trees.


strain 1
WH Human1 China 2019
2019 nCoV WHU02
2019nCoV USA AZ1 202
BR AM L70 CD1733
bat SL CoVZC45
bat SL CoVZXC21
HKU3 3
HKU3 6
HKU3 9
BtCoV2792005
BtCoV2732005
As6526
Rs4237
Rs9401
WIV1
SARS ExoN1 2010
BJ182 12
TW7
NC 004718
HKU 39849
Sin848
BtKY72
0.1



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). The original data we used here were taken from the original SARS-CoV-2 paper Wu et.al, 2020. In the first part of the tutorial (where we used Centrifuge) we simulated a situation where the virus was not in the db and indeed our hits were for "similar" viruses. This is what happened at the beginning of the SARS-CoV-2 pandemic. In the second part of the tutorial (using Genome Detective) the db used included SARS-CoV-2, simulating a scenario where the virus causing the disease/outbreak is not unknown. This is for example similar to what's happening with the monkeypox outbreak.