# Thomson lab –*de novo* assembly SOP

**Emma Thomson 30th April 2021**

## Sample cleaning, trimming and assembly

Before you run your de novo assembly, make a directory to put the results into:

**mkdir DeNovo**

Then soft link your fastq files into that folder as follows:

**ln -s /folder/fastq .**

The current de novo assembly pipeline script on paired raw fastq files should be copied into your DeNovo folder and run as follows:

**cp /home2/HCV2/Uganda/Scripts/De-Novo-scripts/Metavic\_denovo\_assembly\_160121.sh .**

**sh Metavic\_denovo\_assembly\_160121.sh**

This script is based on the **Metavic cleaning and assembly scripts** and is set up to remove low quality reads, to remove adapters, to remove human genome and then to assemble the clean fq files.

Blastx is also carried out both on the raw reads and on the contigs and the results are imported into html files. If you want to take a look at what’s in your samples, open the html files with an internet browser e.g. firefox. You can do this on the server or you can download the html files and open them on your own computer (a lot faster).

The input is **paired fastq** files. To run unpaired reads, the script would require modification.

For optimal *de novo* assembly, it may be best to optimise the kmer settings in this script.

An additional scaffold builder step could be added here to optimise contig assembly but is not currently built in.

To run the script, copy it into a DeNovo/ folder, soft link the fastq files using the ln -s command and run it using sh.

**NB – IMPORTANT WARNING** - we often use an underscore as a delimiter in our scripts. It is really important that before you run the script that you make sure there is enough to distinguish your sequences in the part *before* the delimiter.

e.g. good name: P23-120526\_R1\_001\_etc.fastq – will leave a stem behind of P23-120526 and remove the rest

e.g. bad choice P23\_120526\_R1\_001\_etc.fastq – this will leave behind P23 only.

## Blastx and blastn: HTML reader scripts

We use an automated series of scripts to carry out blastx on the contigs that are generated by *de novo* assembly. Blastx is used first because it is very *sensitive* and is unlikely to miss anything and it can be run quickly. However, it is not *specific* and may you often find false positive hits. Therefore, blastx hits need to be verified by blastn. The html reader scripts take the html output generated by the metavic script (the html files are generated by ktImportBLAST) and create a set of folders that contain organised blastx data. These blastx folders are appended by **\_bx**. Inside the \_bx folders, there are files that include a fasta file of the contigs that match any particular virus. For ease, and to make the subsequent blastn search more efficient, these viruses are organised into HUMAN, Insect, Animal folders etc…

There are some different options/variations for running the html\_reader scripts. It is possible to toggle between local blast for a known virus of interest e.g. HCV and webBLAST for the most up to date blastn.

**STEP 1.** To run the blastx, you need to copy the relevant pipeline file into the DeNovo folder you have created where the \_bx files are located and then run it as follows:

**cp /home2/HCV2/Uganda/Scripts/HTML-reader-scripts/html\_reader\_full\_ET\_new.sh .**

**sh html\_reader\_full\_ET\_new.sh**

You will now have blastx files in your DeNovo folder. Take a look at them by typing:

**ls -d \*\_bx/**

**STEP 2.** To run the blastn, you need to copy the relevant pipeline file into the DeNovo folder you have created where the \_bx files are located as follows:

**/home2/HCV2/Uganda/Scripts/HTML-reader-scripts/** **html\_reader\_human\_blastn\_avi.sh**

Note that there are a few different options for the blastn scripts – you need to chose the right one for you. The differences are to do with what folders will be accessed for blastn and

**OPTION 1**: general html\_reader script e.g. **html\_reader\_human\_blastn\_avi.sh (this version runs a web blastn on \*\_bx/HUMAN/** subdirectories for all human viruses found.

**OPTION 2**: virus specific html reader script e.g. **html\_reader\_hcv\_blastn\_ET2.sh (this version runs a local blastn for HCV)**

**IMPORTANT WARNING:** Do not move any of the metavic generated files in the DeNovo folder as the html\_reader script needs to access these to retrieve the sequence data associated with the contigs.

Key output files from the blastx/blastn html reader files are as follows:

1. **Table\_blastn\_data** – this table lists the top hit for every contig generated by de novo assembly, the relevant sample, the name of the contig and the length of the contig. You can list these as below e.g. for the HUMAN viruses (you can change this folder to Animal, Insect or just \* as required). The length is pretty useful – if it is the virus you want and the right size, you can take this contig and use it for mapping.

**ls -l \*bx/HUMAN/\*/Table\***

1. **Blastn** results – this is the full blastn result for your data – it’s quite a long file

**ls -l \*bx/HUMAN/\*/blastn**

1. **Blastn\_summary** file – this lists all the hits for each contig – it’s a good idea to take a look because occasionally there is a mistake in the ncbi database and the hit is listed as human when it is actually viral
2. You can grep the contig of interest from the **blastx.fa** file in the same folder

grep -A 1 “contig\_name” **blastx.fa**

## Checking for uncategorised viruses

The html script reader files above rely on a database of categorised viruses in the html\_reader\_4 subscript (of the **html\_reader\_full\_ET\_new.sh** script). This needs to be updated regularly and you should check if you have any uncharacterised viruses that have not been assigned to HUMAN, Insect, Animal folers, etc. These viruses appear in the following file that is generated by the **html\_reader\_full\_ET\_new.sh** script automatically:

**Uncharacterised\_viruses\_for\_html\_4**

You should always check if there are viruses in here and these should be characterised and added to the **/home2/HCV2/Uganda/Scripts/HTML-reader-scripts/html\_reader\_4.sh** script

## Validation by mapping

If a whole genome is found by *de novo* assembly, it is usually preferable to use that as the final reference genome for mapping. However, if it isn’t found, it can be much more sensitive to do mapping from the fastq files, using the reference sequence that has been detected by blastn. You can find the accession number in the Table\_blastn\_data that is generated by html\_reader\_5. You can generate a summary of all the viruses found in the \_bx folders that have been confirmed by blastn by using the relevant pathogen selection script – this generates a list of **accession numbers** alone and a list of **accession numbers *and* virusnames.** You need bot of these to proceed to the next stage.

**cp /home2/HCV2/Uganda/Scripts/HTML-reader-scripts/animal\_pathogen\_selection.sh .**

**sh animal\_pathogen\_selection.sh**

Note that there are animal, human and insect versions of the above script (relating to the relevant bx folders). The key outputs are:

**human\_pathogen\_accession\_list**

**human\_pathogen\_names\_and\_accessions\_list**

You should take these outputs and make a new folder for mapping, copy your fastq files using a **soft link** and copy the get\_seq\_with\_accession\_number script

**mkdir Mapping**

**cp human\*list Mapping/**

**ln -s /sourcefolder/\*fastq .**

**cp /home2/HCV2/Uganda/Scripts/DeNovo-scripts/get\_seq\_with\_accession\_number.sh .**

The next step is to use your accession list to download the reference sequences from ncbi using as follows (take a quick look to check the file does contain accession numbers before you run it):

**sh get\_seq\_with\_accession\_number.sh human\_pathogen\_accession\_list**

Next you can download and run the Tanoti script – this will run every .fa fasta file in the directory – so keep it clean! Note – you will need to add .fa to the end of your accession files first e.g. as follows:

**rename ‘s/\.1/\.1\.fa/g’ \*.1**

**rename ‘s/\.2/\.2\.fa/g’ \*.2**

**rename ‘s/\.3/\.3\.fa/g’ \*.3**

The general Tanoti pipeline is as follows: **Tanoti\_batch\_script\_260519.sh.** To run it, copy it into your Mapping folder and run.

**IMPORTANT WARNING** – this script runs ***every*** .fa file against ***every***  fastq file in the folder – so don’t include too many unless you are very sure… Also, it is essential to make sure there will be enough room on the server for this activity.

**cp /home2/HCV2/Uganda/Scripts/Mapping-scripts-2/Tanoti\_batch\_script\_260519.sh .**

**sh Tanoti\_batch\_script\_260519.sh**

One of the reasons that we run every fastq file against every fasta is to check for low level contamination. You can visualise your results by running a nice R script which creates a pdf of the results as follows:

**cp /home2/HCV2/Uganda/Scripts/Mapping-scripts-2/coverage\_table\_avi.sh .**

## Validation by checking the open reading frame

Remember that the open reading frame(s) of your final sequence of interest (if it is a coding part of the genome that you are looking for) should translate. You can check this using **ORF finder** either on the web or on the server.

## Phylogeny

You should check the position of your genome on a phylogenetic tree. If you are comparing to members of a diverse family, it may be necessary to do a protein alignment. It is essential to find the right reference sequences for this by creating a list of these. You should refer to the ICTV website and PubMed for published literature on how to define your virus of interest. Sometimes the whole genome is used for this purpose and sometimes specific genes – refer to the literature and ICTV for the relevant rules for your virus(es) of interest. NB the rules are different for every virus.

## Cross-contamination analysis

We use mapping run as above to check for cross-contamination and have a set of scripts for this – see separate document.