Using the metagenomic pipeline

## Setting up the pipeline

1. Download the pipeline from github as a zip file
   1. The repository for the pipeline can be found on Github as <https://github.com/rach-w/Metagenomic_taxonomy_pipeline> From here you can download as a zip
   2. Alternatively, you can load the git module and clone it from the key with the following command on the hcc:

module load git

git clone git@github.com:rach-w/Metagenomic\_taxonomy\_pipeline.git

1. After successfully cloning/downloading, create a conda environment using the environment.yml file with the following command:

conda env create –f environment.yml

1. Download your host reference genome from the NCBI database
   1. I like to use the ncbi-download module on swan then you can use a simple command:

module load ncbi-download

datasets download genome accession GCF\_016920785.2 --include genome

* 1. Unzip and then make sure to rename this reference sequence without any periods in the name
     1. This messes up the file naming for bowtie host removal steps
     2. I personally like to move the file outside to a set directory for the reference genome which makes it a little easier to enter the path & keep everything organized

1. Download or make sure you have some path to a nt and diamond database
   1. Diamond database is already on shared folder in databases
   2. I downloaded the nt database locally by copying it from the biodata module, but you can also download from online
      1. You can do this with the following commands:

module load biodata

cp $BLAST/nt.\* /nt\_data(or whatever you want to name this)

## Creating a slurm job

1. Create a slurm file via nano with the following command:

nano your\_name.slurm

1. You can use the following template for your slurm job:

#!/bin/bash

#SBATCH --job-name=tax\_pipeline\_pacificus

#SBATCH --mail-user=[rwurachel@gmail.com](mailto:rwurachel@gmail.com)

#SBATCH --mail-type=ALL

#SBATCH --time=0-90:00:00

#SBATCH --output=%x\_%j.out

#SBATCH --error=%x\_%j.err

#SBATCH --nodes=1

#SBATCH --tasks-per-node=32

#SBATCH --mem=250G

#SBATCH --partition=batch

START\_DIR=$(pwd)

HOST\_NAME=$(hostname)

RUN\_DATE=$(date)

echo “Starting working directory: ${START\_DIR}”

echo “Host name: ${HOST\_NAME}”

echo “Run date: ${RUN\_DATE}”

##nextflow

nextflow run main.nf --host\_fasta /your/host/fasta/sequence

--input /your/desired/input/dir

--output /your/desired/output/dir

--local\_blast\_nt /path/to/your/nt/database

--blast\_tax\_dir /same/as/path/to/nt/data/

--local\_diamond /work/crosslab/shared/diamond\_database

--ncbi\_tax\_dir /path/to/ncbi/tax/dir \*this can be blank, pipeline will download

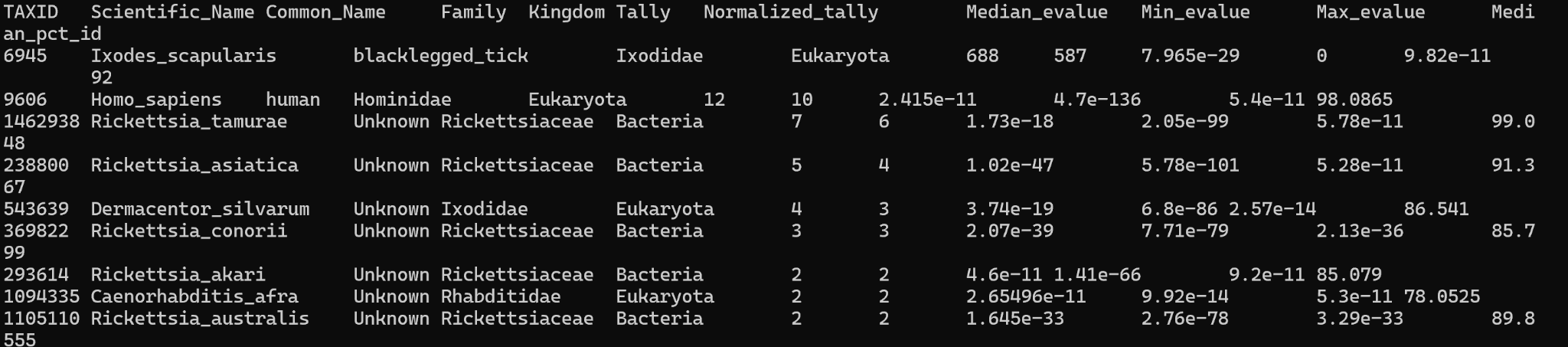
--threads 32 \

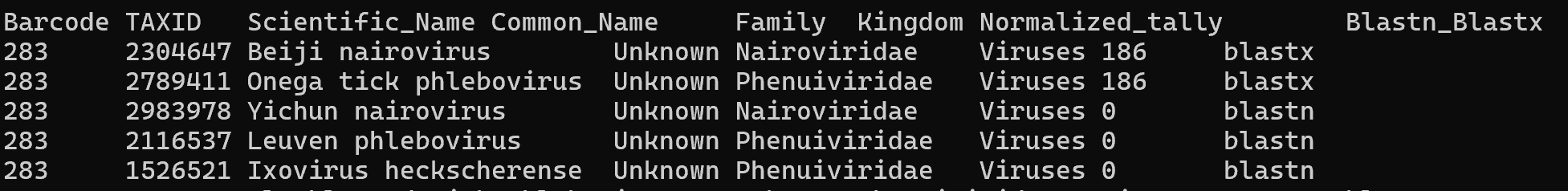
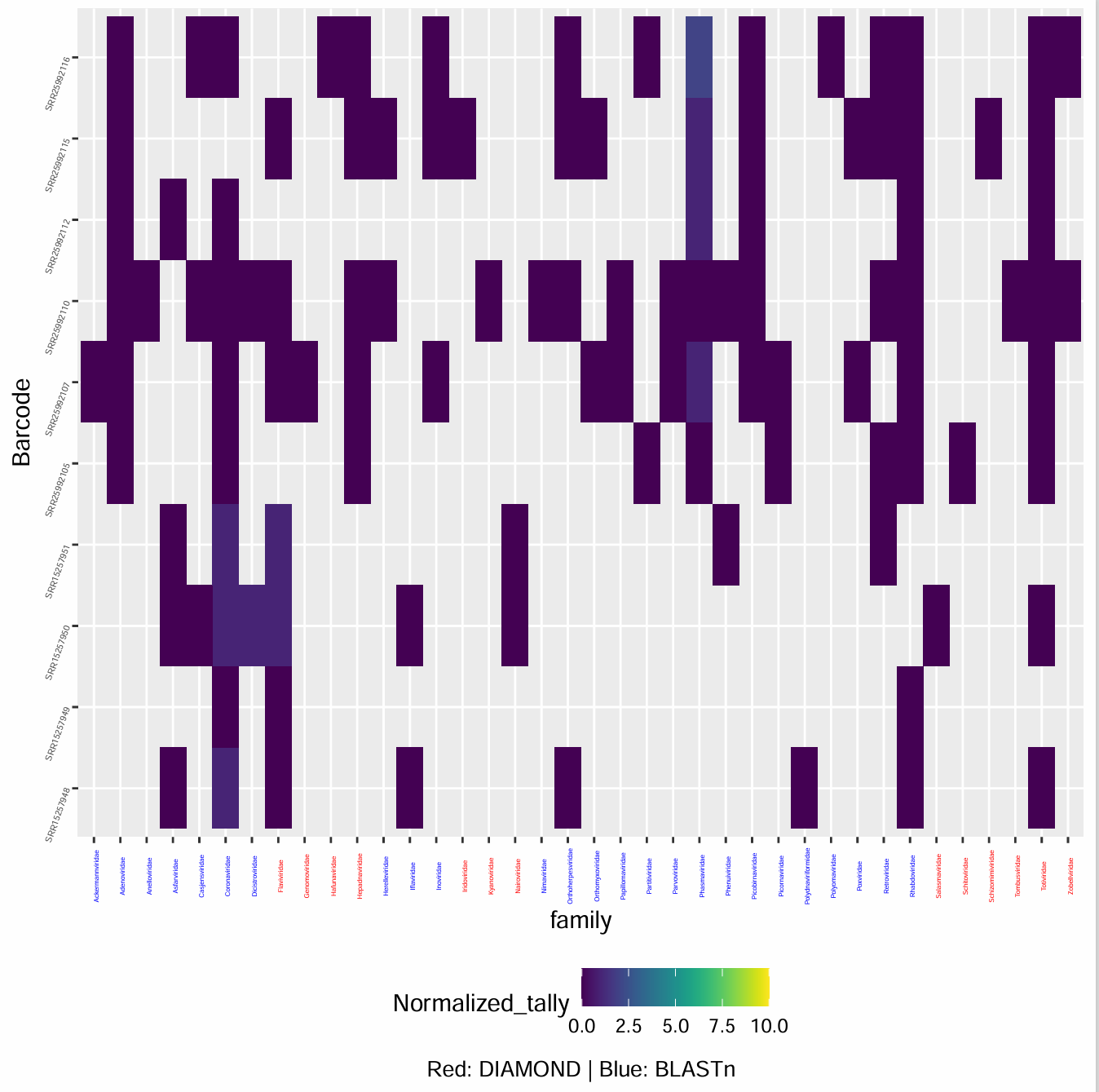
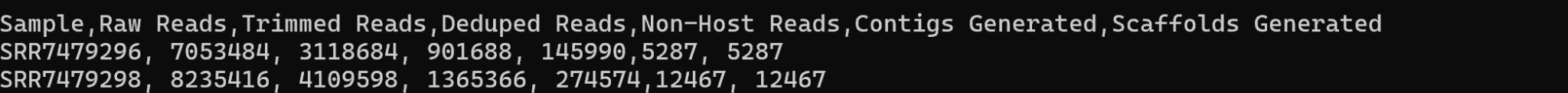
* Some notes about batching your job:
  + BLASTN and BLASTX are both very memory intensive, and for large runs I’ve set memory usage to be 526GB, but you can also choose to set it lower but then run smaller batches
  + Using batch with this large memory will result in a long wait time, so you can use the crosslab node but the memory has to be under 250G
  + Alternatively, using the guest node will go faster and allows for higher memory, so you can specify this with the partition=guest but it might be kicked off mid-run
    - To continue a run after, you can add the –resume flag to your nextflow command so it will pick up from the previous run
* Now once you batch your slurm job you are all good!

## More about the pipeline

### Pipeline outputs

* Tally File
  + 11 columns, with taxid, scientific name, common name, family, kingdom, tally, normalized tally and evalue and pct\_id statistics
  + Normalized tally: Calculated by finding reads per million unique reads
  + Tally # x Total unique reads/1,000,000



* Taxa matrix
  + Tsv file with 8 columns, with barcode, taxid, scientific name, common name, family, kingdom, normalized tally and Blastn\_Blastx
  + Will combine all samples into one file; barcode=sample\_id
  + This tsv file is what is used for the generate heatmap step, and can also be used standalone for any figure generation
  + 
* Heatmap
  + 
  + Generates heatmap with viridis color scheme, can set to either family or species level & names of species/family are in blue or red depending on if they were in the blastn or blastx file
  + May need more adjustments from default settings – can run this script separately or another jupyter notebook to edit output
* Summary.csv
  + Statistics about each sample with reads after each step
  + 
* Virus reads folder has individual fasta sequences of viral species found in each sample
  + These can then be analyzed in geneious if wanted
* Fastqc files for each step are in the corresponding fastqc folders

### Getting extra functionality

* Tally script
* Can filter for a specific kingdom, right now it’s default without filter but you can do -f “Viruses” for example
* More parameters like max eval or tally cutoff
* Distribute fasta by blast
* Also option to do virus only or create all fasta files, or filter certain taxids in a file
* Taxa matrix
* Various options to limit taxids, filter for kingdoms, exclude families, etc