**Downloading gff3 datafiles from CNCB ftp website (individual sequence data)**

-Files are located for download at: <ftp://download.big.ac.cn/GVM/Coronavirus/gff3>

-The individual files are located in subfolders listed as a, b, c …

-To download I use the wget tool on command line, which can be run in a worker node. I download all the gff3 files to a single directory on the server. There are currently 209,445 gff3 files in that single directory. I am not sure if there will ever be issues with too many files in a single folder but for now things are working fine. To download I navigate to the directory that I am storing the files in and run the command below changing the last directory (a,b,c…) to whichever folder I am downloading.

wget -r -l 1 -N -nH -nd <ftp://download.big.ac.cn/GVM/Coronavirus/gff3/a/>

-r – recursive retrieving, needed to download the whole directory

-l 1 – is the number of subdirectories to search in recursive retrieval. Since I am only looking at a single directory I set to 1.

-N – turns on timestamping, checks the timestamp of when files are updated and only updates files that have a newer date. Prevent redundant downloads.

-nH – Does not create directories where downloading data to

-nd – Does not create subdirectories when recursively retrieving

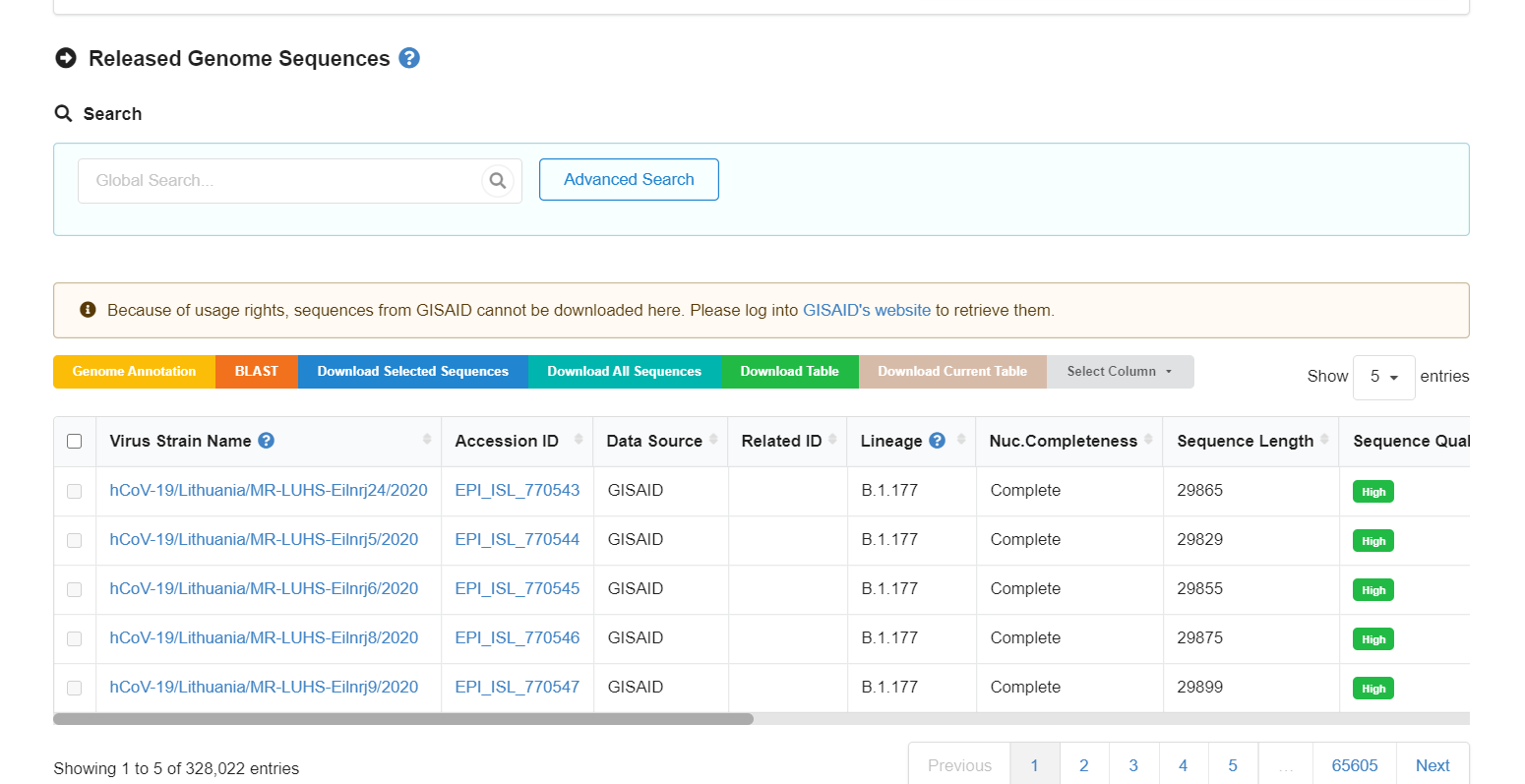
-wget downloading takes some time when there are a lot of files. If there is a large amount of new files it is a good idea to run the download overnight.

**Meta file for gff3 sequence files above, contains date/countries/identification number etc…**

-I download this file directly from the CNCB website. Located at:

<https://bigd.big.ac.cn/ncov/release_genome>

Meta file downloaded from link in red box below:



**Downloading case/death counts per country from John Hopkins**

-Date is located on the Github repository: <https://github.com/CSSEGISandData/COVID-19>

-I just cloned the whole directory onto the Scripps server and use git pull to update the data files. There are two main files that I use in their file structure. There is a time series count for each country and a cumulative country csv. I use the time series to get individual counts per day for each country. The cumulative file I simply use to extract a country population number by using the incident rate that is reported in the cumulative data.

-Population is back calculated through the incident rate number:

population = confirmed\_cases/incident\_rate\_per100k \* 100,000 np.ceil(jh\_data['Confirmed']/jh\_data['Incident\_Rate']\*100000)

The two files are located at:

-Time Series

COVID-19/csse\_covid\_19\_data/csse\_covid\_19\_time\_series/ time\_series\_covid19\_confirmed\_global.csv

COVID-19/csse\_covid\_19\_data/csse\_covid\_19\_time\_series/ time\_series\_covid19\_deaths\_global.csv

-Cumulative for population calculation

COVID-19/csse\_covid\_19\_data/csse\_covid\_19\_daily\_reports

-To update files in Github directory just run:

**git pull**

**Accessing worker nodes with Slurm on new server**

1. First ssh onto any of the login nodes on the server with local port forwarding

**ssh dshak@login02.scripps.edu -L 8956:localhost:8956**

2. Start a tmux session (<https://tmuxcheatsheet.com/> is useful for commands). Used for keeping ssh session alive if you loose connection from the server. Can use screen as well if more familiar with.

**tmux**

3. Allocate computation resource. Specifying number of nodes, compute time and exclusive, which just takes all the resources on a single node. Useful video (<https://asciinema.org/a/366845>)

**salloc --nodes=1 --time=48:00:00 --exclusive**

4. Find job number to attach to. After requesting resources, you are still in login node. Below command will print out information and there should be a field with a job id number.

**export | grep SLURM**

5. Run the job to connect:

**srun --jobid=####(number from above) --pty bash**

6. ssh connect worker node to login node to complete port forwarding.

**ssh dshak@login02.scripps.edu -N -X -f -R 8956:localhost: 8956**

7. Start jupyter lab server:

**jupyter lab --port=8956 --no-browser**

8. copy url from command print out and paste in browser

**Pymol workflow to color protein structures**

1. Start up a worker node.

2. Load the pymol module.

**module load pymol**

3. Run custom pymol script on command line.

**pymol -cq color\_atoms\_single.py**

-c launch in command line mode

-q suppress startup message