Prepare Dataset

#Create Project Directory

cd PATH_TO_PROJECT_FOLDER_SPOT
mkdir -p PATH_TO_PROJECT_FOLDER_SPOT/PROJECT_FOLDER_NAME

#Load raw data files

cd PROJECT_FOLDER_NAME
mkdir 00-RawData

nano sra.sh
CHANGE EMAIL, accession numbers, and folder names and copy/paste into sra.sh####

```
#!/bin/bash
#SBATCH --job-name=sra3            ## job name
#SBATCH -A JRANZ_LAB               ## account to charge
#SBATCH -p standard                ## partition name
#SBATCH -N 1                       ## run on a single node, cant run across multiple
nodes
#SBATCH --ntasks=8                 ## CPUs to use as threads in fasterq-dump command
#SBATCH --tmp=100G                 ## requesting 100 GB local scratch
#SBATCH --constraint=fastscratch   ## requesting nodes with fast scratch in /tmp
#SBATCH --mail-type=fail,end
#SBATCH --mail-user=aekimura@uci.edu

source ~/miniconda3/bin/activate
conda activate PROJECT_NAME

# TMPDIR is created automatically by SLURM
# change to your temp directory assigned by SLURM to your job
cd $TMPDIR

# here we work on just 2 sequences
for f in {447..448}
do
  # generate ID to prefetch, each ID is SRR1196 plus what is contained in $f variable
  ID=SRR1196${f}

  # prefetch SRA file
  prefetch $ID

  # convert sra format to fastq format using requested number of threads (slurm tasks)
  # temp files are written to fastscratch in $TMPDIR with a 100G limit
  fasterq-dump ./$ID/$ID.sra -e $SLURM_NTASKS --temp $TMPDIR --disk-limit-tmp 100G

  # compress resulting fastq files
  gzip $ID*fastq
```

```
done

# move all results to desired location in DFS, directory must exists
mv *fastq.gz  PATH_TO_RAWDATA_FOLDER

conda deactivate
conda deactivate
```
##

#Rename files (move) so they make sense
#There should be a folder for each paired end sample

```
cd 00-RawData
mkdir AC_01
mkdir AC_02
mkdir AC_03
mkdir AL_01
mkdir AL_02
mkdir AL_03

mv SRR7904508_1.fastq.gz AC_01/AC_01_R1.fastq.gz
mv SRR7904508_2.fastq.gz AC_01/AC_01_R2.fastq.gz
mv SRR7904509_1.fastq.gz AC_02/AC_02_R1.fastq.gz
mv SRR7904509_2.fastq.gz AC_02/AC_02_R2.fastq.gz
mv SRR7904510_1.fastq.gz AC_03/AC_03_R1.fastq.gz
mv SRR7904510_2.fastq.gz AC_03/AC_03_R2.fastq.gz
mv SRR7904511_1.fastq.gz AL_01/AL_01_R1.fastq.gz
mv SRR7904511_2.fastq.gz AL_01/AL_01_R2.fastq.gz
mv SRR7904512_1.fastq.gz AL_02/AL_02_R1.fastq.gz
mv SRR7904512_2.fastq.gz AL_02/AL_02_R2.fastq.gz
mv SRR7904513_1.fastq.gz AL_03/AL_03_R1.fastq.gz
mv SRR7904513_2.fastq.gz AL_03/AL_03_R2.fastq.gz

#check to make sure everything was moved correctly
ls -lah */*
```

#Create samples.txt (While in raw data folder)

```
ls >../samples.txt
cat ../samples.txt
```

#Prepare experiment folders

```
cd MAIN_PROJECT_FOLDER (cd ../ if in raw data folder)
mkdir References
```

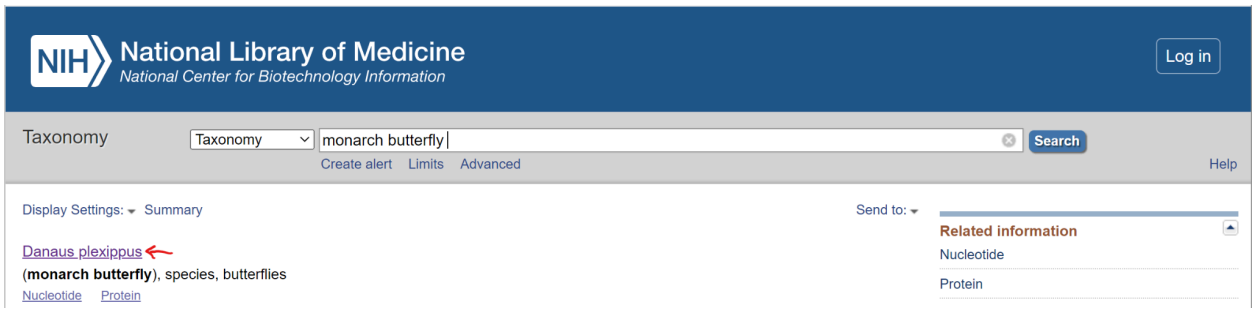
```
mkdir slurmout
mkdir 01-HTS_Preproc
```

Preprocessing Data

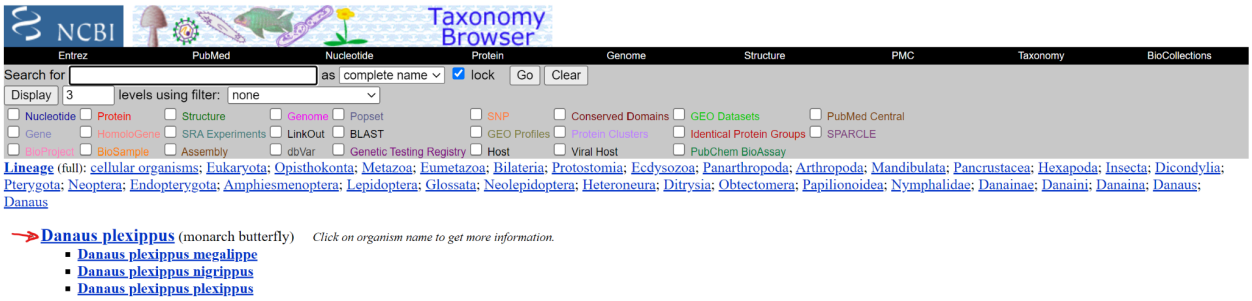
#Download rRNA sequences from NCBI

Go to NCBI. Search “Taxonomy” for “monarch butterfly”

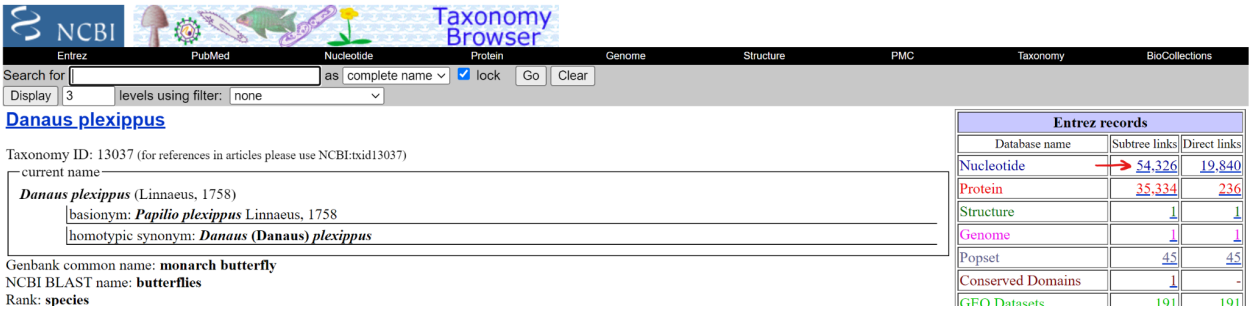
Click on “Danaus plexippus”.



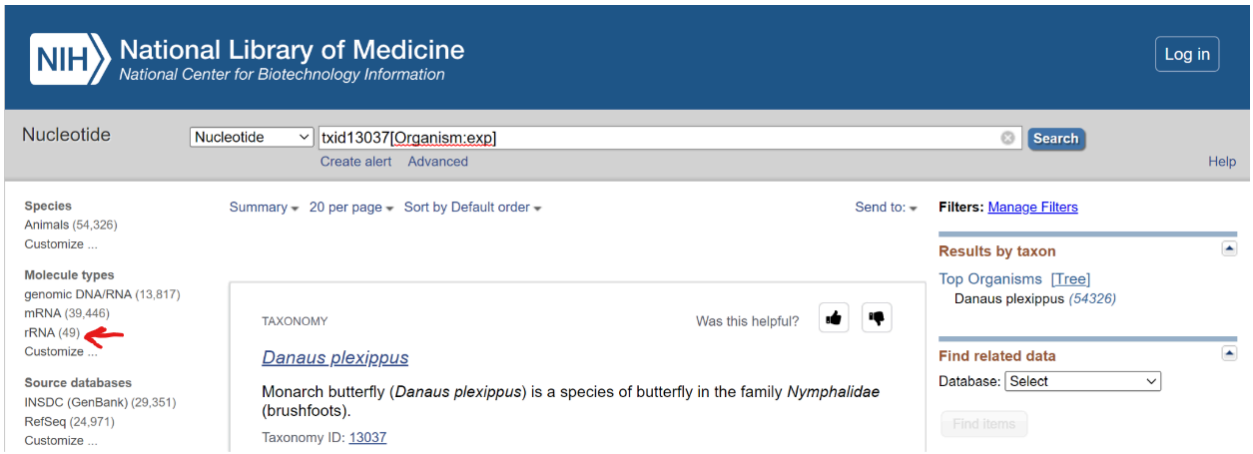
Click on “Danaus plexippus” again.



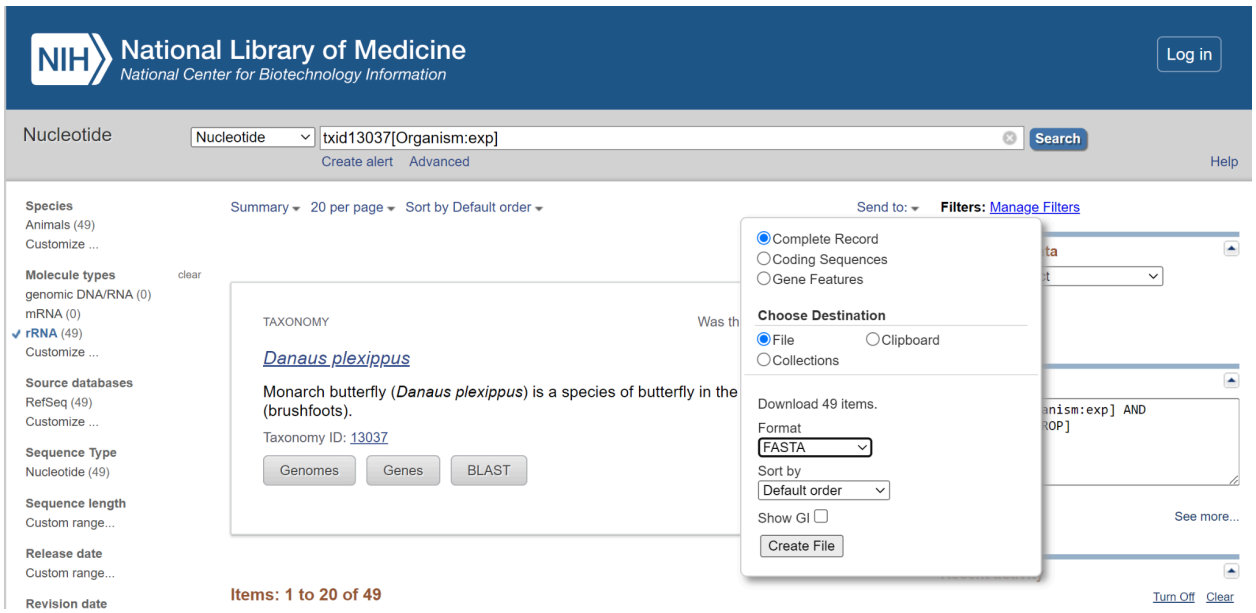
Click on the “Subtree links” for Nucleotide.



On the far left under Molecule types, click “rRNA(49)”



On the right by filters click on “Send to”, choose “File”, choose Format “FASTA”, and click on “Create File”



Save this file to your computer, and rename it to “butterfly_rrna.fasta”
Move the file to “References” directory in your project directory
** scp butterfly_rrna.fasta USER_NAME@hpc3.rcic.uci.edu:/data/homezvol2/hailil2/butterflies/References

1. To transfer a single file `myfile.txt` from your laptop to HPC3 and put it in the directory `/pub/panteater`

   On your laptop, use a Terminal app and descend into the directory where your file is located, then execute the `scp` command using your UCnetID:

```
[user@login-x:~]$ scp myfile.txt panteater@hpc3.rcic.uci.edu:/pub/panteater/myfile.txt
```

#Run HTStream on your project (in your conda environment)

cd PROJECT_DIRECTORY
wget https://ucdavis-bioinformatics-training.github.io/2022-August-RNA-Seq-Analysis/software_scripts/scripts/hts_preproc.slurm
nano hts_preproc.slurm

##########################CHANGE FILE TO MATCH BELOW##########################

```
#!/bin/bash

#SBATCH --job-name=htstream # Job name
#SBATCH -A JRANZ_LAB              ## account to charge
#SBATCH --nodes=1
#SBATCH --ntasks=9
#SBATCH --time=120
#SBATCH --mem=3000 # Memory pool for all cores (see also --mem-per-cpu)
#SBATCH -p standard               ## partition name
#SBATCH --array=1-18
#SBATCH --output=slurmout/htstream_%A_%a.out # File to which STDOUT will be written
#SBATCH --error=slurmout/htstream_%A_%a.err # File to which STDERR will be written
#SBATCH --mail-type=fail,end
#SBATCH --mail-user=aekimura@uci.edu

source ~/miniconda3/bin/activate
conda activate PROJECT_NAME

start=`date +%s`
echo $HOSTNAME
echo "My SLURM_ARRAY_TASK_ID: " $SLURM_ARRAY_TASK_ID

sample=`sed "${SLURM_ARRAY_TASK_ID}q;d" samples.txt`

inpath="00-RawData"
outpath="01-HTS_Preproc"
[[ -d ${outpath} ]] || mkdir ${outpath}
[[ -d ${outpath}/${sample} ]] || mkdir ${outpath}/${sample}

echo "SAMPLE: ${sample}"

#module load htstream/1.3.3

call="hts_Stats -L ${outpath}/${sample}/${sample}.json -N 'initial stats' \
          -1 ${inpath}/${sample}/*R1.fastq.gz \
          -2 ${inpath}/${sample}/*R2.fastq.gz | \
      hts_SeqScreener -A ${outpath}/${sample}/${sample}.json -N 'screen phix' | \
      hts_SeqScreener -A ${outpath}/${sample}/${sample}.json -N 'count the number of rRNA reads'\
          -r -s References/butterfly_rrna.fasta | \
      hts_SuperDeduper -A ${outpath}/${sample}/${sample}.json -N 'remove PCR duplicates' | \
      hts_AdapterTrimmer -A ${outpath}/${sample}/${sample}.json -N 'trim adapters' | \
      hts_PolyATTrim -A ${outpath}/${sample}/${sample}.json -N 'remove polyAT tails' | \
      hts_NTrimmer -A ${outpath}/${sample}/${sample}.json -N 'remove any remaining N characters' | \
      hts_QWindowTrim -A ${outpath}/${sample}/${sample}.json -N 'quality trim the ends of reads' | \
      hts_LengthFilter -A ${outpath}/${sample}/${sample}.json -N 'remove reads < 50bp' \
          -n -m 50 | \
      hts_Stats -A ${outpath}/${sample}/${sample}.json -N 'final stats' \
          -f ${outpath}/${sample}/${sample}"

echo $call
eval $call

end=`date +%s`
runtime=$((end-start))
echo $runtime

conda deactivate
conda deactivate
```

##

```
#Run MultiQC on output files (may not work, just try and look at error output)
#make sure you are in the conda environment for your project
cd PROJECT_DIRECTORY
mkdir -p 02-HTS_multiqc_report
multiqc -i HTSMultiQC-cleaning-report -o 02-HTS_multiqc_report ./01-HTS_Preproc

pip install --upgrade --force-reinstall git+https://github.com/s4hts/MultiQC.git
multiqc -i HTSMultiQC-cleaning-report -o 02-HTS_multiqc_report ./01-HTS_Preproc
```

https://github.com/s4hts/MultiQC/issues/8
https://github.com/ewels/MultiQC/issues/1898

Not working two week before 4/28/23
Package owner working on integration for htstream

Indexing a Genome

#Looking at GTF
(only if necessary)

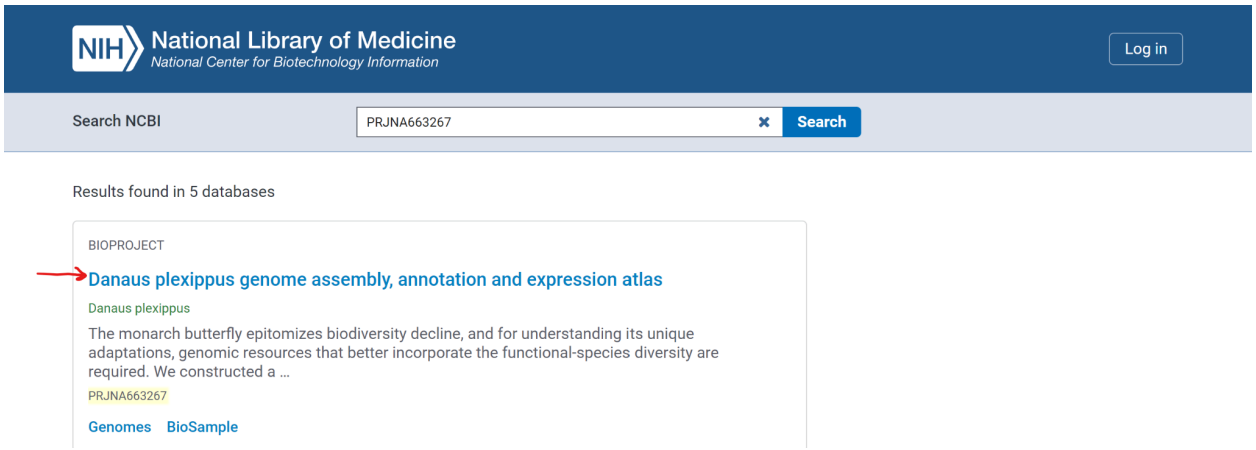
R code:

```
library(GenomicFeatures)      # For loading data from GTF files

# Define the path to the GTF file
gtf_file <- "dmel-all-r6.47.gtf.gz"

# Use the 'read.delim()' function to read in the GTF file
gtf_data <- read.delim(gzfile(gtf_file), header = FALSE, comment.char = "#")

# Separate the 'V9' column into four columns and remove the words
# 'gene_id', 'gene_symbol', 'transcript_id', and 'transcript_symbol'
gtf_data <- gtf_data %>%
  separate(V9, into = c("gene_id", "gene_symbol", "transcript_id", "transcript_symbol"),
           sep = "; ",
           remove = TRUE,
           convert = TRUE) %>%
  mutate(gene_id = gsub("gene_id ", "", gene_id),
         gene_symbol = gsub("gene_symbol ", "", gene_symbol),
         transcript_id = gsub("transcript_id ", "", transcript_id),
         transcript_symbol = gsub("transcript_symbol ", "", transcript_symbol))

# Remove ";" from the end of transcript_symbol
gtf_data$transcript_symbol <- gsub(";$", "", gtf_data$transcript_symbol)
gtf_data$gene_symbol <- gsub(";$", "", gtf_data$gene_symbol)

# Rename columns
colnames(gtf_data) <- c("chromosome", "database", "feature_type", "start", "end",
                        "score", "strand", "phase", "gene_id", "gene_symbol",
                        "transcript_id", "transcript_symbol")
```

#Download the butterfly genome assembly and annotation

We will be using the genome assembly and annotation from Dr. Ranz's paper in Communications Biology: doi: 10.1038/s42003-021-02335-3

For the genome assembly:
Go to NCBI. Search "PRJNA663267" which is the bioproject for this publication. Then open the bioproject.



Scroll down to the Project Data section and click on the assembly link

Project Data:

Resource Name	Number of Links
SEQUENCE DATA	
Nucleotide (total)	24
WGS master	1
SRA Experiments	38
OTHER DATASETS	
BioSample	37
Assembly	1

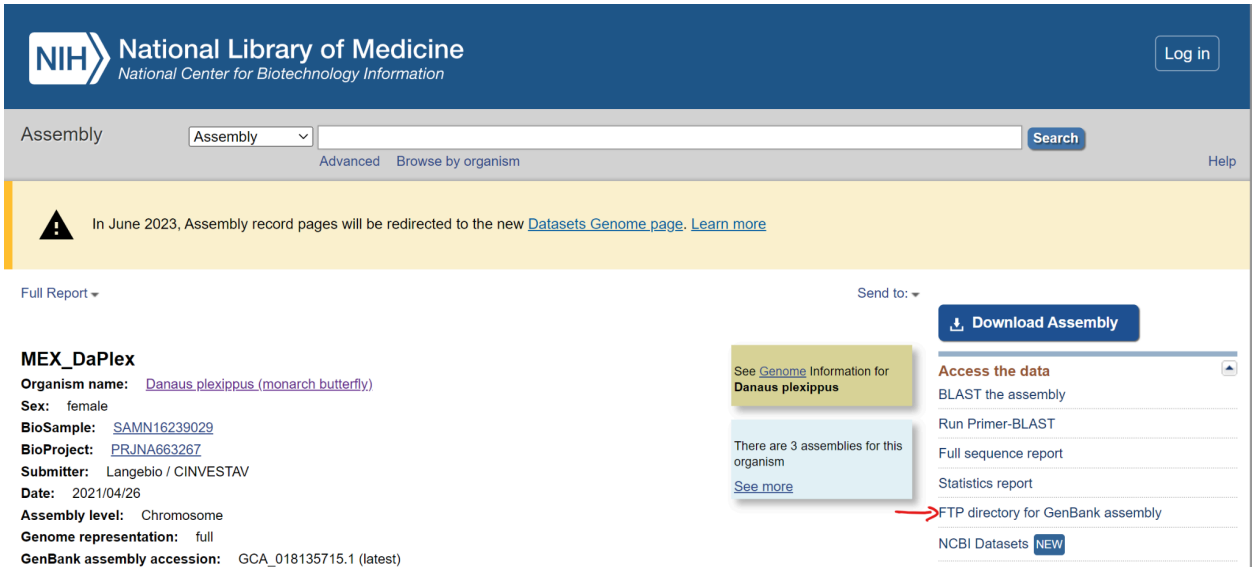
Assembly details: **Download**

Assembly	Level	WGS	Chrs	BioSample	Taxonomy
GCA_018135715.1	Chromosome	JAEQBL000000000	23	SAMN16239029	Danaus plexippus

SRA Data Details

Parameter	Value
Data volume, Gbases	358
Data volume, Tbytes	0.14

Follow the link on the right to the FTP site for this assembly



Download the genome assembly fasta file

Index of /genomes/all/GCA/018/135/715/GCA_018135715.1_MEX_DaPlex

Name	Last modified	Size
Parent Directory		-
GCA_018135715.1_MEX_DaPlex_assembly_structure/	2023-02-05 03:24	-
GCA_018135715.1_MEX_DaPlex_assembly_report.txt	2021-05-23 03:54	6.8K
GCA_018135715.1_MEX_DaPlex_assembly_stats.txt	2023-02-05 03:24	42K
GCA_018135715.1_MEX_DaPlex_feature_count.txt.gz	2021-04-28 12:09	148
GCA_018135715.1_MEX_DaPlex_genomic.fna.gz	2021-04-28 12:09	75M
GCA_018135715.1_MEX_DaPlex_genomic.gbff.gz	2023-02-05 03:24	97M
GCA_018135715.1_MEX_DaPlex_genomic_gaps.txt.gz	2021-04-28 12:09	571
GCA_018135715.1_MEX_DaPlex_wgsmaster.gbff.gz	2023-02-05 03:24	1.7K
README.txt	2020-09-02 16:26	43K
annotation_hashes.txt	2023-02-05 03:24	410
assembly_status.txt	2023-04-24 03:16	14
md5checksums.txt	2023-02-05 03:24	9.9K

For the annotation:
Follow the link: https://zenodo.org/record/4470132#.ZEcTgnbMLD4
Download the file: mxv1.200520.ragoo.rnm.gtf.gz

Move both the files to “References” directory in your project directory
** scp butterfly_rrna.fasta USER_NAME@hpc3.rcic.uci.edu:/data/homezvol2/hailil2/butterflies/References

1. To transfer a single file `myfile.txt` from your laptop to HPC3 and put it in the directory `/pub/panteater`

   On your laptop, use a Terminal app and descend into the directory where your file is located, then execute the `scp` command using your UCnetID:

```
[user@login-x:~]$ scp myfile.txt panteater@hpc3.rcic.uci.edu:/pub/panteater/myfile.txt
```

#Download the STAR module into your conda environment

conda install star

#Index the genome with STAR (in your conda environment)

cd MAIN_PROJECT_DIRECTORY
wget https://raw.githubusercontent.com/ucdavis-bioinformatics-training/2022-August-RNA-Seq-Analysis/master/software_scripts/scripts/star_index.slurm
nano star_index.slurm
#######################CHANGE FILE TO MATCH BELOW#########################

```
#!/bin/bash

#SBATCH --job-name=star_index # Job name
#SBATCH -A JRANZ_LAB              ## account to charge
#SBATCH --nodes=1
#SBATCH --ntasks=16
#SBATCH --time=120
#SBATCH --mem=40000 # Memory pool for all cores (see also --mem-per-cpu)
#SBATCH -p standard               ## partition name
#SBATCH --output=slurmout/star-index_%A.out # File to which STDOUT will be written
#SBATCH --error=slurmout/star-index_%A.err # File to which STDERR will be written
#SBATCH --mail-type=fail,end
#SBATCH --mail-user=aekimura@uci.edu

source ~/miniconda3/bin/activate
conda activate PROJECT_NAME

start=`date +%s`
echo $HOSTNAME

outpath="References"
mkdir -p ${outpath}

cd ${outpath}

#wget https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_mouse/release_M29/GRCm39.primary_assembly.genome.fa.gz
gunzip GCA_018135715.1_MEX_DaPlex_genomic.fna.gz
#FASTA="../GRCm39.primary_assembly.genome.fa"
FASTA="/PATH_TO_REFERENCE_DIRECTORY/mxv1.200520.ragoo.rnm.fa"

#wget
https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_mouse/release_M29/gencode.vM29.primary_assembly.annotation.gtf.
gz
gunzip mxv1.200520.ragoo.rnm.gtf.gz
#GTF="../gencode.vM29.primary_assembly.annotation.gtf"
GTF="/PATH_TO_REFERENCE_DIRECTORY/mxv1.200520.ragoo.rnm.gtf"

mkdir star.overlap100.MEX_DaPlex
cd star.overlap100.MEX_DaPlex

#module load star/2.7.10a

call="STAR
     --runThreadN 8 \
     --runMode genomeGenerate \
     --genomeDir . \
     --genomeFastaFiles ${FASTA} \
     --sjdbGTFfile ${GTF} \
     --sjdbOverhang 100 \
     --genomeSAindexNbases 12"

echo $call
eval $call

end=`date +%s`
runtime=$((end-start))
echo $runtime

conda deactivate
conda deactivate
```

##

Change of database for the **fasta file**
https://drive.google.com/file/d/17HAwSbsPiZSvMDSabNRcvnbiFrF_-XK8/view

mkdir: cannot create directory ‘star.overlap100.MEX_DaPlex’: File exists
!!!!! WARNING: --genomeSAindexNbases 14 is too large for the genome size=245173502, which may cause seg-fault at the mapping step. Re-run genome generation with recommended --genomeSAindexNbases 12

Aligning with STAR

#Run STAR on the project (in your conda environment)

cd MAIN_PROJECT_DIRECTORY
mkdir 02-STAR_alignment
wget https://raw.githubusercontent.com/ucdavis-bioinformatics-training/2022-August-RNA-Seq-Analysis/master/software_scripts/scripts/star.slurm
nano star.slurm
##########################CHANGE FILE TO MATCH BELOW##########################

```
#!/bin/bash

#SBATCH --job-name=star # Job name
#SBATCH -A JRANZ_LAB              ## account to charge
#SBATCH --nodes=1
#SBATCH --ntasks=8
#SBATCH --time=60
#SBATCH --mem=32000 # Memory pool for all cores (see also --mem-per-cpu)
#SBATCH -p standard               ## partition name
#SBATCH --array=1-6
#SBATCH --output=slurmout/star_%A_%a.out # File to which STDOUT will be written
#SBATCH --error=slurmout/star_%A_%a.err # File to which STDERR will be written
#SBATCH --mail-type=fail,end
#SBATCH --mail-user=aekimura@uci.edu

source ~/miniconda3/bin/activate
conda activate PROJECT_NAME

start=`date +%s`
echo $HOSTNAME
echo "My SLURM_ARRAY_TASK_ID: " $SLURM_ARRAY_TASK_ID

sample=`sed "${SLURM_ARRAY_TASK_ID}q;d" samples.txt`
REF="References/star.overlap100.MEX_DaPlex"

outpath='02-STAR_alignment'
[[ -d ${outpath} ]] || mkdir ${outpath}
[[ -d ${outpath}/${sample} ]] || mkdir ${outpath}/${sample}

echo "SAMPLE: ${sample}"

#module load star

call="STAR
      --runThreadN ${SLURM_NTASKS} \
      --genomeDir $REF \
      --outSAMtype BAM SortedByCoordinate \
      --readFilesCommand zcat \
      --readFilesIn 01-HTS_Preproc/${sample}/${sample}_R1.fastq.gz 01-HTS_Preproc/${sample}/${sample}_R2.fastq.gz \
      --quantMode GeneCounts \
      --outFileNamePrefix ${outpath}/${sample}/${sample}_ \
      > ${outpath}/${sample}/${sample}-STAR.stdout 2> ${outpath}/${sample}/${sample}-STAR.stderr"
```

```
echo $call
eval $call

end=`date +%s`
runtime=$((end-start))
echo $runtime

conda deactivate
conda deactivate
```

##

#Check to see if STAR worked correctly

cd MAIN_PROJECT_DIRECTORY
wget https://raw.githubusercontent.com/ucdavis-bioinformatics-training/2022-August-RNA-Seq-Analysis/master/software_scripts/scripts/star_stats.sh
nano star_stats.sh
#####To run, input the following command, don't do sbatch######
bash star_stats.sh

Move the "summary_star_alignments.txt" file to your computer to read

2. To transfer a single file `j-123.fa` from HPC3 to your laptop

   On your laptop, use a Terminal app and descend into the directory where you want to transfer the file, then execute the `scp` command using your UCnetID.

```
[user@login-x:~]$ scp panteater@hpc3.rcic.uci.edu:/pub/panteater/project1/j-123.fa j-123.fa
```

Generating counts tables

```
cd MAIN_PROJECT_DIRECTORY

mkdir 03-Counts
mkdir 03-Counts/tmp
for sample in `cat samples.txt`; do \
   echo ${sample}
   cat 02-STAR_alignment/${sample}/${sample}_ReadsPerGene.out.tab | tail -n +5 | cut -f4 > 03-Counts/tmp/${sample}.count
done

tail -n +5 02-STAR_alignment/First_folder_name/first_file_name_ReadsPerGene.out.tab | cut -f1 > 03-Counts/tmp/geneids.txt

paste 03-Counts/tmp/geneids.txt 03-Counts/tmp/*.count > 03-Counts/tmp/tmp.out

cat <(cat samples.txt | sort | paste -s) 03-Counts/tmp/tmp.out > 03-Counts/rnaseq_counts.txt
rm -rf 03-Counts/tmp
```

Move the “rnaseq_counts” file to your computer to analyze in R!

2. To transfer a single file `j-123.fa` from HPC3 to your laptop

   On your laptop, use a Terminal app and descend into the directory where you want to transfer the file, then execute the `scp` command using your UCnetID.

```
[user@login-x:~]$ scp panteater@hpc3.rcic.uci.edu:/pub/panteater/project1/j-123.fa j-123.fa
```

/share/crsp/lab/jranz/share/SequencingData/Founders_RNAseq