#2.2: Creating an RNA-seq analysis pipeline

SLURM scripting and processing

Using bioinformatics tools to process RNA-seq data

Transferring files between CHPC and personal computer

IGV

Access the slides and files here:

https://github.com/j-berg/bioinformatics_bootcamp

Introduction to SLURM

- Schedule jobs to be run on high-performance compute nodes
- Keeps the usage fair for everyone
  - If you've run a bunch of jobs recently, you are put at the end of the queue
- Submit a SLURM script (a modified bash script)

Scratch Directory

- Temporary file storage
- Unlimited space
- Be sure to clear out files you don’t need!
- If using human/mouse samples, need to build your index here due to storage constraints

- Location:

$ mkdir /scratch/general/lustre/uNID

SLURM job header

```
#!/bin/bash                                                          <-tell script where to find bash
#SBATCH --time=72:00:00                                              <- script time-out
#SBATCH --nodes=1                                                    <- how many nodes to use
#SBATCH -o /uufs/chpc.utah.edu/common/home/uNID/slurmjob-%j          <- Where to write the job log
#SBATCH --partition=notchpeak                                        <- Where to run the job

source /uufs/chpc.utah.edu/common/home/uNID/miniconda3/etc/profile.d/conda.sh
source activate class                                                <- Access conda environment
--------------------------------------------------------------------------------
```

% is a magic character -- %j will auto-fill SLURM job ID

Create a workflow for genome indexing

```
#!/bin/bash
#SBATCH --time=72:00:00
#SBATCH --nodes=1
#SBATCH -o /uufs/chpc.utah.edu/common/home/u       /slurmjob-%j
#SBATCH --partition=notchpeak

source /uufs/chpc.utah.edu/common/home/u       /miniconda3/etc/profile.d/conda.sh
source activate class

SCRUSER=/scratch/general/lustre/u
REF=/scratch/general/lustre/u       /yeast_index
GTF=$REF/Saccharomyces_cerevisiae.R64-1-1.100.gtf

# initialize reference folder in scratch directory
mkdir -p $REF
mv ~/reference_yeast/* $REF

# Isolate FASTA files
mkdir -p $REF/fastas
mv $REF/*.fa $REF/fastas

# Generate STAR index
STAR --runMode genomeGenerate --genomeDir $REF/genome --genomeFastaFiles $REF/fastas --sjdbGTFfile $REF/$GTF --runThreadN
 32 --sjdbOverhang 50
```

Run a workflow for genome indexing

```
(class) [u       @notchpeak2 ~]$ vim make_yeast_index.sh
(class) [u       @notchpeak2 ~]$ sbatch make_yeast_index.sh
```

```
Every 1.0s: squeue -u u                                          Sun Jun  7 13:50:22 2020

             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           1398079 notchpeak make_yea u         R       3:04      1 notch012
```

Creating an analysis workflow

```
#!/bin/bash
#SBATCH --time=72:00:00
#SBATCH --nodes=1
#SBATCH -o /uufs/chpc.utah.edu/common/home/u       /slurmjob-%j
#SBATCH --partition=notchpeak

source /uufs/chpc.utah.edu/common/home/u       /miniconda3/etc/profile.d/conda.sh
source activate class

# Initialize output directory
SCRDIR=/scratch/general/lustre/$USER/$SLURM_JOBID
mkdir -p $SCRDIR

INPUT=/uufs/chpc.utah.edu/common/home/u       /seq_files
REF=/scratch/general/lustre/u       /yeast_index
GTF=$REF/Saccharomyces_cerevisiae.R64-1-1.100.gtf
FILES=(SRR1166442 SRR1166443 SRR1166444 SRR1166445 SRR1166446 SRR1166447)
OUTPUT=/uufs/chpc.utah.edu/common/home/u       /seq_output

# Move raw data
mkdir -p $SCRDIR/input
mkdir -p $SCRDIR/output
cp $INPUT/* $SCRDIR/input

# Organize output
mkdir -p $SCRDIR/output/preprocess
mkdir -p $SCRDIR/output/alignment
mkdir -p $SCRDIR/output/postprocess
mkdir -p $SCRDIR/output/count
mkdir -p $SCRDIR/output/qc

mkdir -p $OUTPUT

cd $SCRDIR/.
```

Creating an analysis workflow

```
# Loop through all files
for FILE in ${FILES[@]}; do

        echo "Processing ${FILE}"
        echo "Preprocessing..."
        # Preprocess
        fastp --thread 20 -l 30 -q 28 \
                -i $SCRDIR/input/${FILE}.fastq \
                -o $SCRDIR/output/preprocess/${FILE}.fastq \
                -j $SCRDIR/output/preprocess/${FILE}.json \
                -h $SCRDIR/output/preprocess/${FILE}.html

        echo "Preprocessing QC..."
        # Perform quality control on pre-processed FASTQ file
        fastqc -q $SCRDIR/output/preprocess/${FILE}.fastq
                -o $SCRDIR/output/qc/${FILE}

        echo "Aligning..."
        # Align
        STAR --runThreadN 20 --sjdbOverhang 50 \
                --outSAMunmapped Within --outSAMtype BAM Unsorted --quantMode TranscriptomeSAM \
                --genomeDir $REF/genome \
                --sjdbGTFfile $GTF \
                --readFilesIn $SCRDIR/output/preprocess/${FILE}.fastq \
                --outFileNamePrefix $SCRDIR/output/alignment/${FILE} #will end in _Aligned.bam

        echo "Postprocessing..."
        # Sort and Index
        samtools sort --threads 20 \
                -o $SCRDIR/output/postprocess/${FILE}_sorted.bam \
                $SCRDIR/output/alignment/${FILE}_Aligned.bam
        samtools index -@ 20 \
                $SCRDIR/output/postprocess/${FILE}_sorted.bam

        echo "Counting..."
        # Count
        htseq-count -q -f bam -m intersection-nonempty -t exon -i gene_id -r pos -s no \
                $SCRDIR/output/postprocess/${FILE}_sorted.bam \
                $GTF > $SCRDIR/output/count/${FILE}.tsv; done
```

Creating an analysis workflow

```
# Summarize QC
multiqc $SCRDIR/output

# Clean-up
mv $SCRDIR/output/postprocess/*_sorted.bam $OUTPUT
mv $SCRDIR/output/postprocess/*_sorted.bam.bai $OUTPUT
mv $SCRDIR/output/count/*.tsv $OUTPUT
mv $SCRDIR/output/qc/*.html $OUTPUT
mv $SCRDIR/output/*.html $OUTPUT

rm -rf $SCRDIR
```

Helpful SLURM commands

Start a slurmjob:

```
$ sbatch jobid
```

Cancel a slurmjob:

```
$ scancel jobid
```

See your jobs in queue:

```
$ squeue -u uNID
```

Live update of queue:

```
$ watch -n1 squeue -u uNID
```

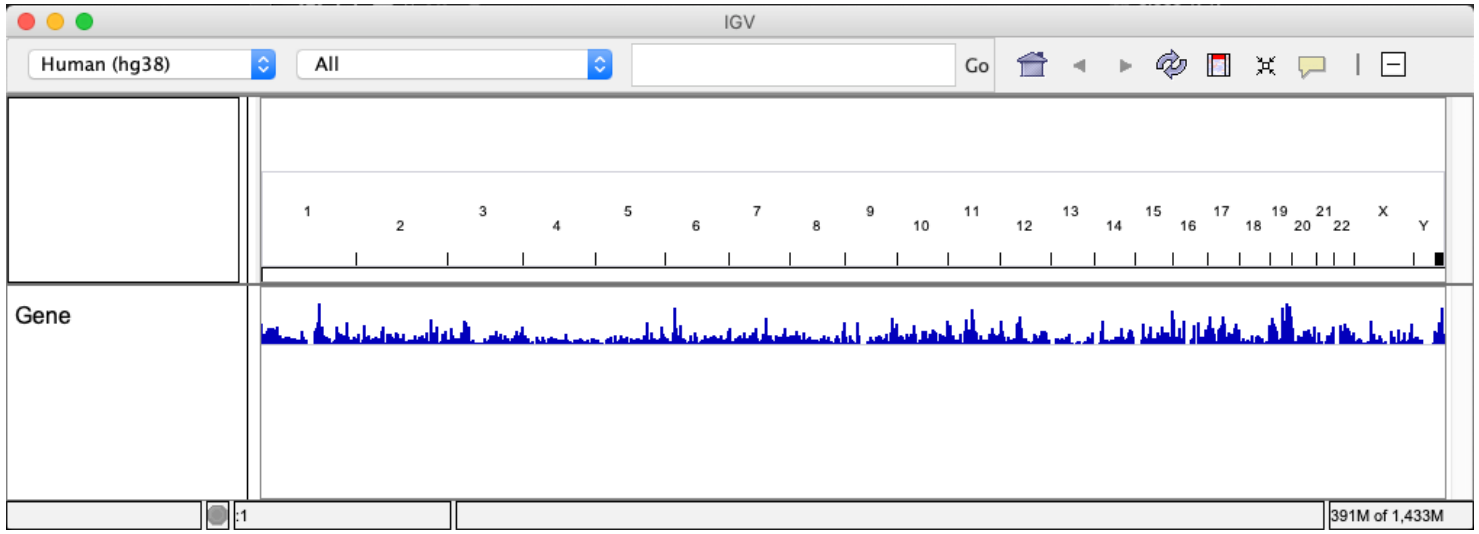
Transferring files to your personal computer

- From your personal computer/terminal

```
$ scp uNID@notchpeak.chpc.utah.edu:~/path/to/file.bam ./
$ scp uNID@notchpeak.chpc.utah.edu:~/path/to/file.bam.bai ./
```

Visualizing read pile-ups with IGV

- Download IGV:
  - https://software.broadinstitute.org/software/igv/download
- Drag and drop BAM file into viewer



Homework

- For the dataset you previously downloaded, download the appropriate reference files for that model organism
- Generate a genome index and store in the Scratch directory
- Create and run a script that processes each of the files you downloaded
- Transfer one of the alignment files to your personal computer and open in IGV. Find a gene whose transcripts (isoforms) seem to be differentially expressed.