Worksheet 4.3 - Mapping reads using HISAT2
Authors: Mary Allen & Daniel Ramirez

HISAT2 manual: https://ccb.jhu.edu/software/hisat2/manual.shtml
Samtools manual: http://www.htslib.org/doc/samtools.html

Username: Screenshots show 'daramirez', though you will see your own username!

1. Using an appropriate terminal, log on to the cluster to use **hisat2**:
    a. Use ***pwd*** to make sure you know where you are and ***ls*** to make sure you know what is in this directory.

```
[daramirez@ip-172-31-12-93 ~]$  pwd
/Users/daramirez
[daramirez@ip-172-31-12-93 ~]$  ls -lh
total 36K
drwxr-xr-x 2 daramirez daramirez 6.0K Jul  8 11:06 Desktop
drwxr-xr-x 2 daramirez daramirez 6.0K Jul  8 11:06 Documents
drwxr-xr-x 2 daramirez daramirez 6.0K Jul  8 11:06 Downloads
drwxrwxr-x 3 daramirez daramirez 6.0K Jul  8 16:43 igv
drwxr-xr-x 2 daramirez daramirez 6.0K Jul  8 11:06 Music
drwxr-xr-x 2 daramirez daramirez 6.0K Jul  8 11:06 Pictures
drwxr-xr-x 2 daramirez daramirez 6.0K Jul  8 11:06 Public
drwxr-xr-x 2 daramirez daramirez 6.0K Jul  8 11:06 Templates
drwxr-xr-x 2 daramirez daramirez 6.0K Jul  8 11:06 Videos
```

    b. Change the working directory (***cd***) to your own scratch directory.

```
[daramirez@ip-172-31-12-93 ~]$  cd /scratch/Users/daramirez/
[daramirez@ip-172-31-12-93 daramirez]$  ls -lh
total 16K
drwxrwxr-x 2 daramirez daramirez 6.0K Jul  8 16:55 eofiles
drwxrwxr-x 2 daramirez daramirez 6.0K Jul  8 16:55 fastQC
drwxrwxr-x 2 daramirez daramirez 6.0K Jul  8 16:55 sbatch
drwxrwxr-x 2 daramirez daramirez 6.0K Jul  8 16:55 trimmomatic
```

2. Make a new directory/folder (***mkdir***) named hisat2.

```
[daramirez@ip-172-31-12-93 daramirez]$  mkdir hisat2
[daramirez@ip-172-31-12-93 daramirez]$  ls -lh
total 20K
drwxrwxr-x 2 daramirez daramirez 6.0K Jul  8 16:55 eofiles
drwxrwxr-x 2 daramirez daramirez 6.0K Jul  8 16:55 fastQC
drwxrwxr-x 2 daramirez daramirez 6.0K Jul  8 16:56 hisat2
drwxrwxr-x 2 daramirez daramirez 6.0K Jul  8 16:55 sbatch
drwxrwxr-x 2 daramirez daramirez 6.0K Jul  8 16:55 trimmomatic
```

    This directory that will contain the results from hisat2. The error and output files generated by your batch script jobs will be stored in "eofiles". The batch script that you will create will live in the "sbatch" directory.

3. Go check the fastq data files in the following public directory using ***cd*** and ***ls***: /scratch/Workshop/SR2019. In there, there are several folders containing fastq files that have all been aligned using hisat2; from ATAC-seq, to ChIP-seq, and RNA-seq. In this example, we will map/align sequencing data from a ChIP-seq experiment from a human cell line. To make this example run quick enough for teaching purposes, a subsample of the whole ChIP-seq file has been produced corresponding to some sequencing reads from chromosome 1 only. We will work with the file

"SRR5855054_chr1.trimmed.fastq" which has been already been trimmed, and lives in /scratch/Workshop/SR2019/4_qc/trimmomatic.

```
[daramirez@ip-172-31-12-93 trimmomatic]$  cd /scratch/Workshop/SR2019/4_qc/trimmomatic/
[daramirez@ip-172-31-12-93 trimmomatic]$  ls -lh
total 113M
drwxr-xr-x 2 magruca magruca 6.0K Jul  8 13:51 fastqc
-rwxr-xr-x 1 magruca magruca  89M Jul  8 13:51 SRR5855054_chr1.trimmed.fastq
-rwxr-xr-x 1 magruca magruca  25M Jul  8 13:51 trimlog
```

4. Find the script batch template "template.sbatch" in the directory: /scratch/Workshop/SR2019/scripts

```
-rwxr--r-- 1 joru1876 dowelldegrp  456 Jul  6 16:21 10_intersect_all
-rwxr--r-- 1 joru1876 dowelldegrp  405 Jul  6 16:21 10_intersect_all.sh
-rwxr-xr-x 1 sahu0957 dowelldegrp 1.4K Jul  5 21:20 4_fastqc.sbatch
-rwxr-xr-x 1 sahu0957 dowelldegrp 1.8K Jul  5 21:20 4_trimmomatic.sbatch
-rw-r--r-- 1 magr0763 dowelldegrp 1.9K Jul  5 11:39 chr1.sbatch
-rw-r--r-- 1 magr0763 dowelldegrp 2.5K Jul  8 13:46 dastk.sbatch
-rw-r--r-- 1 magr0763 dowelldegrp 2.2K Jul  8 13:46 macs2.sbatch
-rwxr-xr-x 1 magr0763 dowelldegrp 4.9M Jul  5 11:18 sr2018_all_scripts.tar.gz
-rwxr-xr-x 1 magr0763 dowelldegrp 1.3K Jul  8 13:56 template.sbatch
```

5. Copy the script batch "template.sbatch" that you just looked at to your sbatch directory /scratch/Users/<YOUR_USERNAME>/sbatch" and change its name to "mapping.sbatch" (***mv <original name> <new name>***).

```
[daramirez@ip-172-31-12-93 ~]$  cd /scratch/Users/daramirez/sbatch/
[daramirez@ip-172-31-12-93 sbatch]$  ls -lh
total 4.0K
-rwxr-xr-x 1 daramirez daramirez 1.3K Jul  8 17:06 template.sbatch
[daramirez@ip-172-31-12-93 sbatch]$  mv template.sbatch mapping.sbatch
[daramirez@ip-172-31-12-93 sbatch]$  ls -lh
total 4.0K
-rwxr-xr-x 1 daramirez daramirez 1.3K Jul  8 17:06 mapping.sbatch
```

6. Complete the new "mapping.sbatch" file with the right content to run hisat2. (hint: transition to insert mode by pressing ***i*** if using vim.

```
#!/bin/bash
#SBATCH --job-name=<JOB-NAME>                       # Job name
#SBATCH --mail-type=ALL                             # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=<EMAIL>                         # Where to send mail
#SBATCH --nodes=<n>                                 # Number of nodes requested
#SBATCH --ntasks=<n>                                # Number of CPUs (processor cores/tasks)
#SBATCH --mem=<n>gb                                 # Memory limit
#SBATCH --time=<00:00:00>                           # Time limit hrs:min:sec
#SBATCH --partition=compute                         # Partition/queue requested on server
#SBATCH --output=/scratch/Users/<USERNAME>/eofiles/<JOB-NAME>.%j.out       # Standard output
#SBATCH --error=/scratch/Users/<USERNAME>/eofiles/<JOB-NAME>.%j.err        # Standard error log

### Displays the job context
echo Job: $SLURM_JOB_NAME with ID $SLURM_JOB_ID
echo Running on host `hostname`
echo Job started at `date +"%T %a %d %b %Y"`
echo Directory is `pwd`
echo Using $SLURM_NTASKS processors across $SLURM_NNODES nodes

### Assigns path variables
INPUT_DIRECTORY=<PATH_TO_INPUT_FILE>
OUTPUT_DIRECTORY=<PATH_TO_OUTPUT_FILE>

### Loads modules
<MODULES_TO_LOAD>

### <SOFTWARE SPECIFICS>

echo Job finished at `date +"%T %a %d %b %Y"`
~
~
"mapping.sbatch" 29L, 1281C
```

```
#SBATCH --job-name=<JOB-NAME>                    # Job name
#SBATCH --mail-type=ALL                          # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=<EMAIL>                      # Where to send mail
#SBATCH --nodes=<n>                              # Number of nodes requested
#SBATCH --ntasks=<n>                             # Number of CPUs (processor cores/tasks)
#SBATCH --mem=<n>gb                              # Memory limit
#SBATCH --time=<00:00:00>                        # Time limit hrs:min:sec
#SBATCH --partition=compute                      # Partition/queue requested on server
#SBATCH --output=/scratch/Users/<USERNAME>/eofiles/<JOB-NAME>.%j.out       # Standard output
#SBATCH --error=/scratch/Users/<USERNAME>/eofiles/<JOB-NAME>.%j.err        # Standard error log
```

a. Change the name of the script batch from <JOB-NAME> to something more useful, such as "mapping".
b. Replace <EMAIL> with your own email address to which you want to receive any notifications.
c. Replace <USERNAME> with your own username to complete the path directory to where to store the error and output files.
d. Complete the following fields: nnodes, ntasks, mem and time. Hisat2 can use multiple processors per input file. So, 1 node, 4 tasks/processors/CPUs, 5 Gb for memory and 5 minutes for wall-time should be enough.

```
#!/bin/bash
#SBATCH --job-name=mapping                       # Job name
#SBATCH --mail-type=ALL                          # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=dara6367@colorado.edu        # Where to send mail
#SBATCH --nodes=1                                # Number of nodes requested
#SBATCH --ntasks=4                               # Number of CPUs (processor cores/tasks)
#SBATCH --mem=5gb                                # Memory limit
#SBATCH --time=00:05:00                          # Time limit hrs:min:sec
#SBATCH --partition=compute                      # Partition/queue requested on server
#SBATCH --output=/scratch/Users/daramirez/eofiles/%x.%j.out       # Standard output
#SBATCH --error=/scratch/Users/daramirez/eofiles/%x.%j.err        # Standard error log
```

e. Specify the path to the input file "`SRR5855054_chr1.trimmed.fastq`" as the value of the variable "INPUT_DIRECTORY". Also specify the path that leads to the "hisat2" directory you created earlier in your scratch directory as the value of the variable "OUTPUT_DIRECTORY".

```
### Assigns path variables
INPUT_DIRECTORY=/scratch/Workshop/SR2019/4_qc/trimmomatic
OUTPUT_DIRECTORY=/scratch/Users/daramirez/hisat2
```

f. Assign the required modules necessary to run this batch script job with both hisat2 and samtools. To look for the correct bowtie2 and samtools modules, exit vim by saving all changes (press ***ESC*** and type ***:wq!***), and in the terminal, list all available modules on the computer cluster that contain the word "hisat2" and "samtools" in them. Do this with the command ***module spider <string>*** and look for the ones for hisat2 and samtools.

```
[daramirez@ip-172-31-12-93 sbatch]$  module spider hisat2

----------------------------------------------------------------------------
  hisat2: hisat2/2.1.0
----------------------------------------------------------------------------
    Description:
      No Description Given

    This module can be loaded directly: module load hisat2/2.1.0
```

```
[daramirez@ip-172-31-12-93 sbatch]$  module spider samtools

----------------------------------------------------------
  samtools:
----------------------------------------------------------
    Description:
      No Description Given

     Versions:
        samtools/1.3.1
        samtools/1.8

----------------------------------------------------------
  For detailed information about a specific "samtools" module
 load the modules) use the module's full name.
  For example:

     $ module spider samtools/1.8
----------------------------------------------------------
```

Using vim, add “module load hisat2/2.1.0” and “module load samtools/1.8” in the file “mapping.sbatch” in the section “<MODULES_TO_LOAD>”.

```
### Loads modules
module load hisat2/2.1.0
module load samtools/1.8
```

g. The last edit you need to do is the actual block of text that specifies how to run hisat2 and a couple of samtools commands needed to obtain a file ready for visualization using the Integrative Genomics Viewer (IGV).

   1) The syntax to use **hisat2** for single-end reads is as follows:
   hisat2 [options] -x <genome_index> -U <input_fastq> **>** <output_sam>

Do not forget to specify the full path to all the files, including the human genome index files. Use the variables that you created earlier to make things easier. You could decide type the whole hisat2 command in a single line, as shown here below:

```
### <SOFTWARE SPECIFICS>

hisat2 --threads 4 --new-summary --very-sensitive --no-spliced-ali
gnment -x /scratch/Workshop/hg38/HISAT2/genome -U $INPUT_DIRECTORY
/SRR5855054_chr1.trimmed.fastq > $OUTPUT_DIRECTORY/SRR5855054_chr1
.trimmed.sam
```

But that is very hard to read. You could instead break up the command onto many lines using the character \ at the end of every line. These \ characters are ignored by the computer, but will help you identify each part of the command more easily.

```
### <SOFTWARE SPECIFICS>

hisat2 \
--threads 4 \
--new-summary \
--very-sensitive \
--no-spliced-alignment \
-x /scratch/Workshop/hg38/HISAT2/genome \
-U $INPUT_DIRECTORY/SRR5855054_chr1.trimmed.fastq \
> $OUTPUT_DIRECTORY/SRR5855054_chr1.trimmed.sam
```

2) The syntax to use **samtools view** is as follows:
samtools view [options] <output_bam> **>** <input.sam>

3) The syntax to use **samtools sort** is as follows:
samtools sort [options] <input_bam> **>** <output_sorted.bam>

4) The syntax to use **samtools index** is as follows:
samtools index <input_sorted.bam> **>** <output_sorted.bam.bai>

```
samtools view \
-@ 4 \
-Sb \
$OUTPUT_DIRECTORY/SRR5855054_chr1.trimmed.sam \
> $OUTPUT_DIRECTORY/SRR5855054_chr1.trimmed.bam

samtools sort \
-@ 4 \
$OUTPUT_DIRECTORY/SRR5855054_chr1.trimmed.bam \
> $OUTPUT_DIRECTORY/SRR5855054_chr1.trimmed.sorted.bam

samtools index \
$OUTPUT_DIRECTORY/SRR5855054_chr1.trimmed.sorted.bam \
> $OUTPUT_DIRECTORY/SRR5855054_chr1.trimmed.sorted.bam.bai
```

So, we will go from having an empty template, to having a complete hisat2 & samtools block of commands and a complete batch script.

```
#!/bin/bash
#SBATCH --job-name=mapping                          # Job name
#SBATCH --mail-type=ALL                             # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=dara6367@colorado.edu           # Where to send mail
#SBATCH --nodes=1                                   # Number of nodes requested
#SBATCH --ntasks=4                                  # Number of CPUs (processor cores/tasks)
#SBATCH --mem=5gb                                   # Memory limit
#SBATCH --time=00:05:00                             # Time limit hrs:min:sec
#SBATCH --partition=compute                         # Partition/queue requested on server
#SBATCH --output=/scratch/Users/daramirez/eofiles/%x.%j.out       # Standard output
#SBATCH --error=/scratch/Users/daramirez/eofiles/%x.%j.err        # Standard error log

### Displays the job context
echo Job: $SLURM_JOB_NAME with ID $SLURM_JOB_ID
echo Running on host `hostname`
echo Job started at `date +"%T %a %d %b %Y"`
echo Directory is `pwd`
echo Using $SLURM_NTASKS processors across $SLURM_NNODES nodes

### Assigns path variables
INPUT_DIRECTORY=/scratch/Workshop/SR2019/4_qc/trimmomatic
OUTPUT_DIRECTORY=/scratch/Users/daramirez/hisat2

### Loads modules
module load hisat2/2.1.0
module load samtools/1.8

### <SOFTWARE SPECIFICS>

hisat2 \
--threads 4 \
--new-summary \
--very-sensitive \
--no-spliced-alignment \
-x /scratch/Workshop/hg38/HISAT2/genome \
-U $INPUT_DIRECTORY/SRR5855054_chr1.trimmed.fastq \
> $OUTPUT_DIRECTORY/SRR5855054_chr1.trimmed.sam

samtools view \
-@ 4 \
-Sb \
$OUTPUT_DIRECTORY/SRR5855054_chr1.trimmed.sam \
> $OUTPUT_DIRECTORY/SRR5855054_chr1.trimmed.bam

samtools sort \
-@ 4 \
$OUTPUT_DIRECTORY/SRR5855054_chr1.trimmed.bam \
> $OUTPUT_DIRECTORY/SRR5855054_chr1.trimmed.sorted.bam

samtools index \
$OUTPUT_DIRECTORY/SRR5855054_chr1.trimmed.sorted.bam \
> $OUTPUT_DIRECTORY/SRR5855054_chr1.trimmed.sorted.bam.bai

echo Job finished at `date +"%T %a %d %b %Y"`
```

Save all changes to the “mapping.sbatch” file and exit vim.

7. Now that the batch script is ready, submit it to the job manager SLURM to begin processing the ChIP-seq sequencing data. In the terminal, while located in the “sbatch” directory where “mapping.sbatch” lives, type ***sbatch <sbatch file>***. The job manager will give you a job number. Once submitted, you can check on the status of jobs by typing ***squeue -u username***.

```
[daramirez@ip-172-31-12-93 sbatch]$  sbatch mapping.sbatch
Submitted batch job 7
[daramirez@ip-172-31-12-93 sbatch]$  squeue -u daramirez
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                 7   compute  mapping daramire PD       0:00      1 (Priority)
```

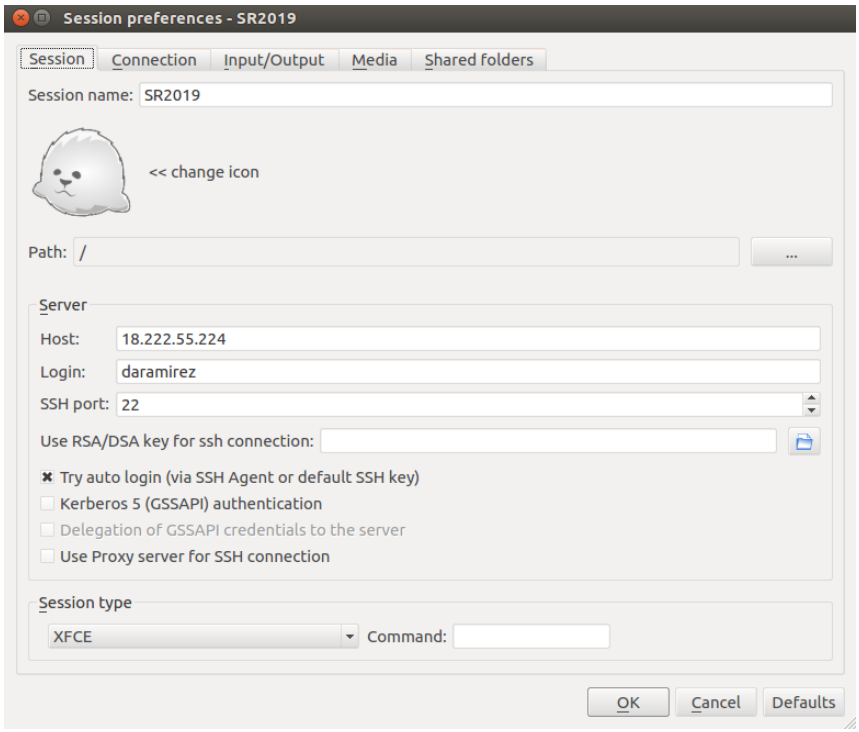
8. Move to the “eofiles” directory and open the error and output files. If your job failed, here is where you should go to figure out what went wrong. If your job succeeded, you can see in the “.err” file the hisat2 alignment report.

```
[daramirez@ip-172-31-12-93 sbatch]$  cd ../eofiles/
[daramirez@ip-172-31-12-93 eofiles]$  ls -lh
total 8.0K
-rw-r--r-- 1 daramirez daramirez 240 Jul  9 14:43 mapping.3435100.err
-rw-r--r-- 1 daramirez daramirez 235 Jul  9 14:43 mapping.3435100.out
[daramirez@ip-172-31-12-93 eofiles]$  more  mapping.3435100.out mapping.3435100.err
::::::::::::::
mapping.3435100.out
::::::::::::::
Job: mapping with ID 3435100
Running on host fijinode-48
Job started at 14:35:03 Tue 09 Jul 2019
Directory is /scratch/Users/dara6367/data/sread2019test/sbatch
Using 4 processors across 1 nodes
Job finished at 14:35:37 Tue 09 Jul 2019
::::::::::::::
mapping.3435100.err
::::::::::::::
HISAT2 summary stats:
        Total reads: 786063
                Aligned 0 time: 25462 (3.24%)
                Aligned 1 time: 600485 (76.39%)
                Aligned >1 times: 160116 (20.37%)
        Overall alignment rate: 96.76%
[bam_sort_core] merging from 0 files and 4 in-memory blocks...
```

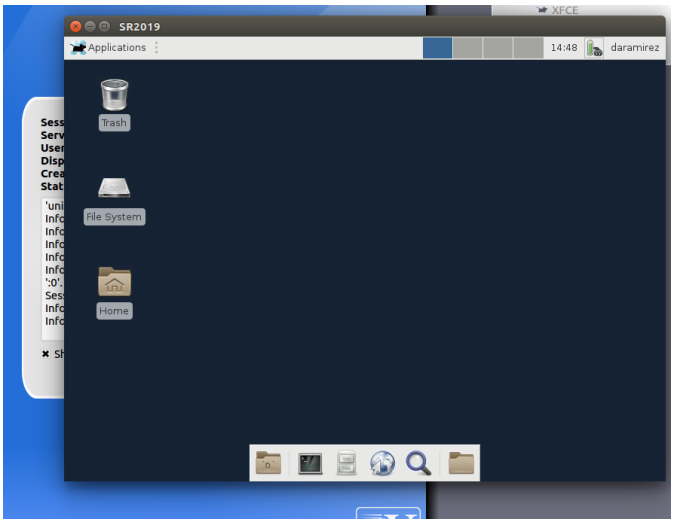
9. Check the “hisat2” directory. There should be four files: a sam file, a bam file, a sorted bam file, and a sorted bam index file.

```
[daramirez@ip-172-31-12-93 eofiles]$  cd ../hisat2/
[daramirez@ip-172-31-12-93 hisat2]$  ls -lh
total 354M
-rw-r--r-- 1 daramirez daramirez  53M Jul  9 14:42 SRR5855054_chr1.trimmed.bam
-rw-r--r-- 1 daramirez daramirez 246M Jul  9 14:42 SRR5855054_chr1.trimmed.sam
-rw-r--r-- 1 daramirez daramirez  53M Jul  9 14:42 SRR5855054_chr1.trimmed.sorted.bam
-rw-r--r-- 1 daramirez daramirez 3.0M Jul  9 14:42 SRR5855054_chr1.trimmed.sorted.bam.bai
```

10. The sorted.bam and bai files are the two necessary files for visualization of the data using IGV. Open “X2Go”. Log in onto a new session window. If you have not configured your session, then you should configure it now. Name your session with a meaningful name in the section “Session name”. In host, type the corresponding server name that you want to connect to, for this class type “18.222.55.224”. Type your GitHub username in “Login”. Select the option “Try auto login (via SSH Agent or default SSH key”. Change “Session type” to “XFCE”. Do not change anything else. Save changes of the new session by clicking “OK”.

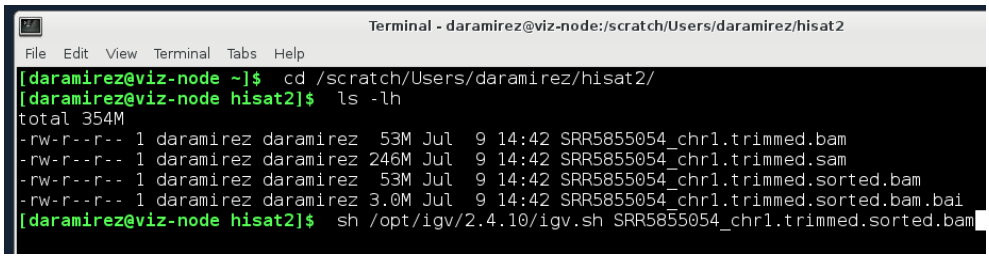


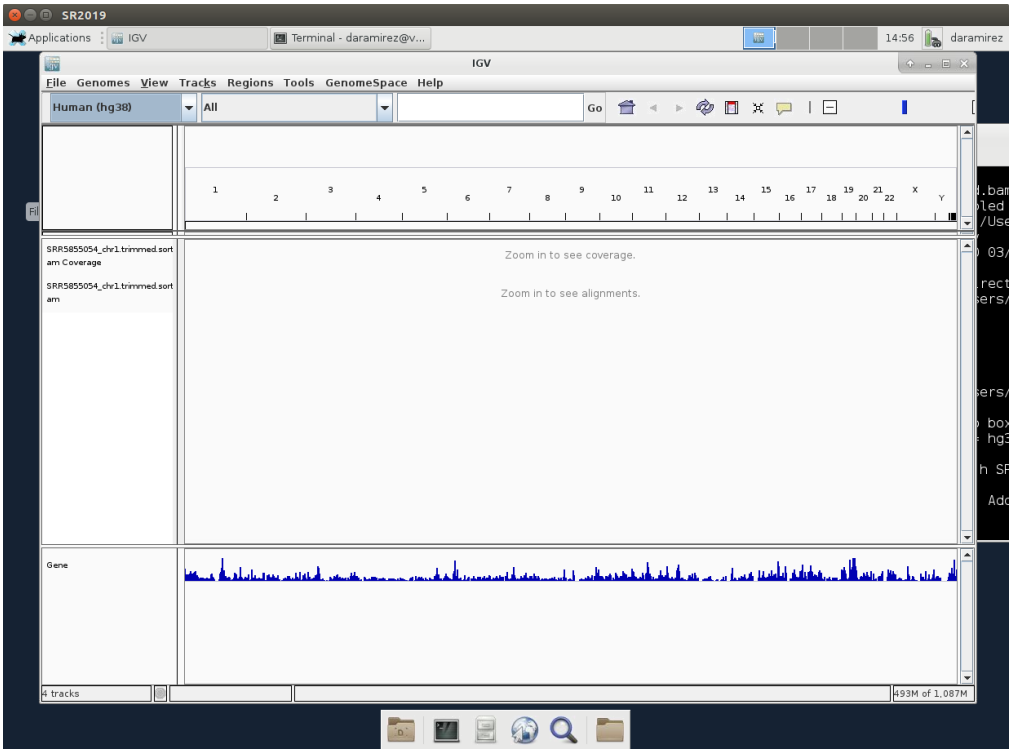
11. Click on the created session box on the right, and select "Yes" if asked if you trust the host key. If successfully connected, a new window will appear. This is the cluster node that you will use to visualize your data using IGV.



12. Increase the size of the window so that IGV can be displayed completely. Open the terminal emulator icon located in the bottom bar of the new window. You can navigate to all your files and directories that you have created so far using the same commands you have learned. Change directory to where your bam files are located. To open IGV along with the bam file, type the following command:

```
sh /opt/igv/2.4.10/igv.sh SRR5855054_chr1.trimmed.sorted.bam
```





13. Finally, customize IGV as you please (font size, rename track, track height, track color, color alignments by condition, etc.). Zoom in onto your favorite chr1 locus.