

Day 4 Worksheet – Read mapping and visualization

Author: Qing Yang (qing.yang@colorado.edu)

HISAT2 manual: <https://ccb.jhu.edu/software/hisat2/manual.shtml>

Samtools manual: <http://www.htslib.org/doc/samtools.html>

IGV manual: <https://software.broadinstitute.org/software/igv/UserGuide>

1. Make sure you have the following files in your `/day4/trimmomatic/` directory:

```
26M Jul 13 14:31 chr21Eric_repA.RNA.end1.trimmed.fastq
  0 Jul 13 14:31 chr21Eric_repA.RNA.end1.unpaired.fastq
26M Jul 13 14:31 chr21Eric_repA.RNA.end2.trimmed.fastq
  0 Jul 13 14:31 chr21Eric_repA.RNA.end2.unpaired.fastq
```

2. Create new directory (`mkdir`) named `hisat2`, under `/day4/`, which will be the output directory for mapped reads.

```
-bash-4.2$ mkdir hisat2
mkdir: created directory 'hisat2'
-bash-4.2$ ls -lsh
total 1.0K
  0 drwxrwsr-x+ 2 qiya9811 dowelldegrp 0 Jul 13 14:24 fastq/
  0 drwxrwsr-x+ 2 qiya9811 dowelldegrp 0 Jul 13 14:33 fastqc/
  0 drwxrwsr-x+ 2 qiya9811 dowelldegrp 0 Jul 13 14:33 hisat2/
512 drwxrwsr-x+ 2 qiya9811 dowelldegrp 1 Jul 13 14:27 scripts/
512 drwxrwsr-x+ 2 qiya9811 dowelldegrp 4 Jul 13 14:31 trimmomatic/
```

3. `cd` into scripts directory and copy (`cp`) the “template.sbatch” job script file and change its name to “mapping.sbatch”.

```
-bash-4.2$ cd scripts/
-bash-4.2$ cp template.sbatch mapping.sbatch
'template.sbatch' -> 'mapping.sbatch'
-bash-4.2$ ls -l
total 3.0K
-rwxrwxr-x+ 1 qiya9811 dowelldegrp 1.3K Jul 13 14:36 mapping.sbatch*
-rwxrwxr-x+ 1 qiya9811 dowelldegrp 1.3K Jul 13 14:27 template.sbatch*
```

4. Edit the new “mapping.sbatch” using the text editor `vim`. First, edit the SBATCH configuration to meet the needs of read mapping:

- a. Change the name of the script batch from to something more useful, such as “hisat2_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, 8 tasks/processors/CPU, 2 Gb for memory and 90 minutes for wall-time should be enough.

```
#!/bin/bash
#SBATCH --job-name=hisat2_mapping          # Job name
#SBATCH --mail-type=ALL                    # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=qing.yang@colorado.edu # Where to send mail
#SBATCH --nodes=1                          # Number of nodes requested
#SBATCH --ntasks=8                         # Number of CPUs (processor cores/tasks)
#SBATCH --mem=2gb                           # Memory limit
#SBATCH --time=<01:30:00>                   # 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
```

5. Next, assign path variables. In this case, we will specify two directories, both under **DATADIR**. **TRIM** stores the directory path to trimmed reads. **HISAT2** stores the directory path to output mapped reads.

```
### Assigns path variables

DATADIR=/scratch/Users/<USERNAME>/day4/
HISAT2=${DATADIR}/hisat2
TRIM=${DATADIR}/trimmomatic
```

6. Next, load the modules/software needed for mapping reads and file conversion:

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

7. And finally, specify the read mapping and file conversion commands. Note that 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:

NOTE: genome index is located at **/scratch/sread/data_files/genome/hg38/HISAT2/genome**

```
### <SOFTWARE SPECIFICS>

## Map trimmed reads to reference genome

hisat2 --very-fast -x /scratch/Shares/public/genomes/hisatfiles/hg38/HISAT2/genome \
-1 ${TRIM}/chr21Eric_repA.RNA.end1.trimmed.fastq \
-2 ${TRIM}/chr21Eric_repA.RNA.end2.trimmed.fastq \
> ${HISAT2}/chr21Eric_repA.RNA.sam \
2> ${HISAT2}/chr21Eric_repA.hisat2_maptstats.txt

## Convert mapped reads to sorted bam file
### convert SAM to BAM
samtools view -@ 8 -bS -o ${HISAT2}/chr21Eric_repA.RNA.bam \
${HISAT2}/chr21Eric_repA.RNA.sam

### sort bam file
samtools sort -@ 8 ${HISAT2}/chr21Eric_repA.RNA.bam \
> ${HISAT2}/chr21Eric_repA.RNA.sorted.bam

### index sorted bam file
samtools index ${HISAT2}/chr21Eric_repA.RNA.sorted.bam \
${HISAT2}/chr21Eric_repA.RNA.sorted.bam.bai
```

8. Before you close vim, make sure to save your edits by press Ese button to exit insertion mode, then type in `:wq` to save and quit vim.

9. Now that the job script is complete, submit the job by type in `sbatch` command. While waiting for the job to execute, you can check the job status using the command `squeue -u <USERNAME>`:

```
-bash-4.2$ sbatch mapping.sbatch
Submitted batch job 7730124
-bash-4.2$ squeue -u qiya9811
      JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
      7730124      short hisat2_m qiya9811  R           0:07        1 fjinode-12
```

10. Finally, check the output directory `/day4/hisat2/` there should be 5 different files:

```
-bash-4.2$ ls -lsh
total 172M
1.0K -rw-rw-r--+ 1 qiya9811 dowelldegrp 613 Jul 13 16:33 chr21Eric_repA.hisat2_maptstats.txt
24M  -rw-rw-r--+ 1 qiya9811 dowelldegrp 24M Jul 13 16:33 chr21Eric_repA.RNA.bam
127M -rw-rw-r--+ 1 qiya9811 dowelldegrp 127M Jul 13 16:33 chr21Eric_repA.RNA.sam
19M  -rw-rw-r--+ 1 qiya9811 dowelldegrp 19M Jul 13 16:33 chr21Eric_repA.RNA.sorted.bam
1.7M -rw-rw-r--+ 1 qiya9811 dowelldegrp 1.7M Jul 13 16:33 chr21Eric_repA.RNA.sorted.bam.bai
```

11. To visualize the mapped reads using IGV, we will need to connect to X2Go client. 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 “**3.15.250.159**”.
- 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**”.

Server

Host: 3.15.250.159

Login: qyang13 Your Github Username

SSH port: 22

Use RSA/DSA key for ssh connection:

Try auto login (via SSH Agent or default SSH key)

Kerberos 5 (GSSAPI) authentication

Delegation of GSSAPI credentials to the server

Use Proxy server for SSH connection

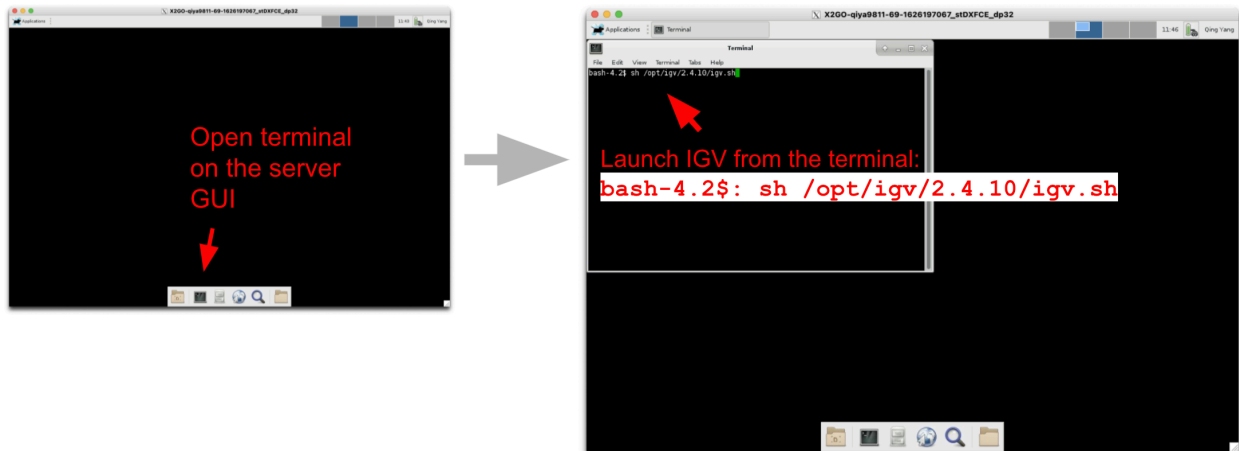
Session type

Run in X2GoKDrive (experimental)

XFCE Command:

12. 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.

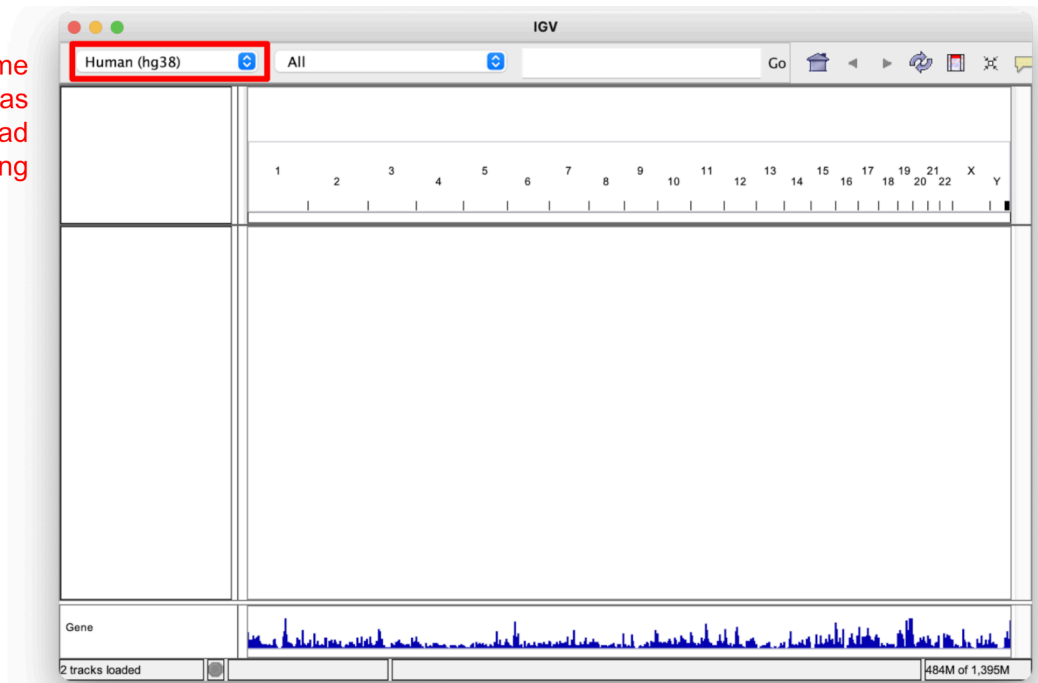
Increase the size of the window so that IGV can be displayed completely. Open the **terminal** icon located in the bottom bar of the new window. You can navigate to all your files and directories that you have created so far using the same commands you have learned. To open IGV type the command **sh /opt/igv/2.4.10/igv.sh**:



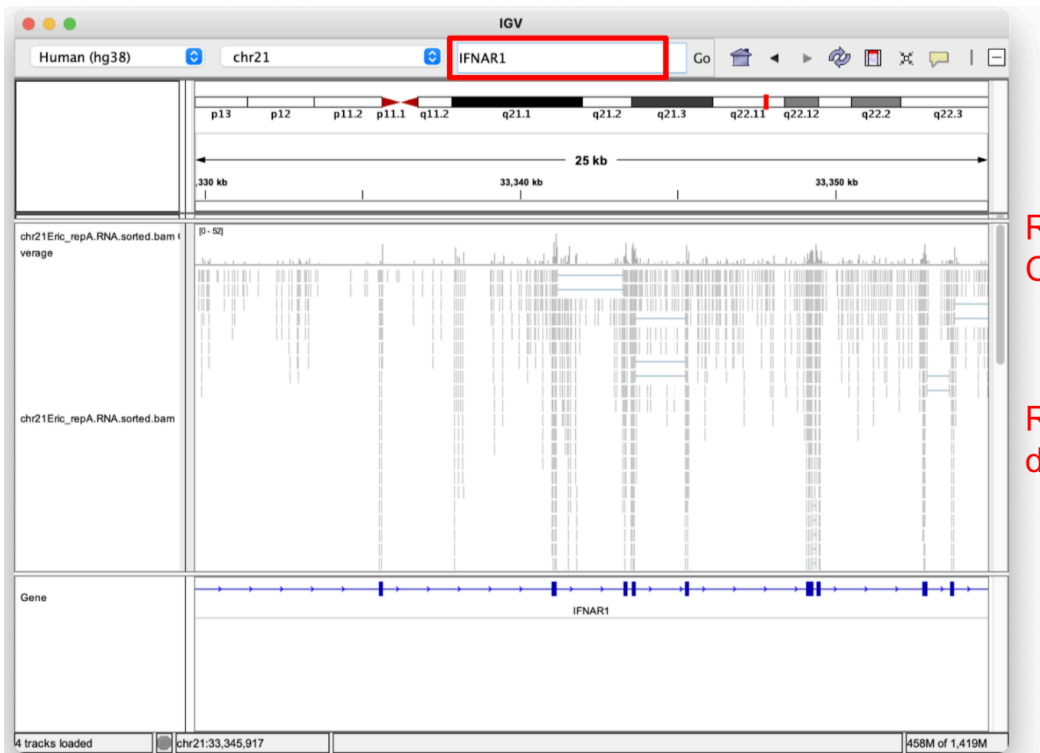
13. Before you load your bam files, make sure the genome is consistent with the reference genome that was used for read mapping. In this case, human hg38 genome is used. In the case if you need to load your custom genome, refer to the IGV manual page:

<https://software.broadinstitute.org/software/igv/LoadGenome>:

Select the same genome used as reference for read mapping



14. Finally, load the bam files we just created using HISAT2 and samtools by clicking on “**File**” > “**Load From File**”, then choose the desired bam files in your HISAT2 output directory. To save time, we only mapped reads from chromosome 21 to the human genome, so to visualize the mapped reads, make sure to zoom into specific loci on chromosome 21. For example, the IFNAR1 gene is shown in the screenshot below. You can zoom in by typing in the gene name in the genome coordinate box, and by adjusting using the +/- button at the top right corner. Here, you should see the mapped reads summarized in both coverage and alignment formats:



Read
Coverage

Read mapping
details