# Variant Calling Workshop

Chris Fields

PowerPoint by Casey Hanson
Edited by Gio Madrigal & Roberto Cucalón Tamayo

# Introduction

In this lab, we will do the following:

1. Perform variant calling analysis on the IGB biocluster.

2. Visualize our results on the desktop using the Integrative Genomics Viewer (IGV) tool.

# Start the VM

- Follow instructions for starting VM (This is the Remote Desktop software).

- The instructions are different for UIUC and Mayo participants.

- Find the instructions for this on the course website under Lab Set-up:

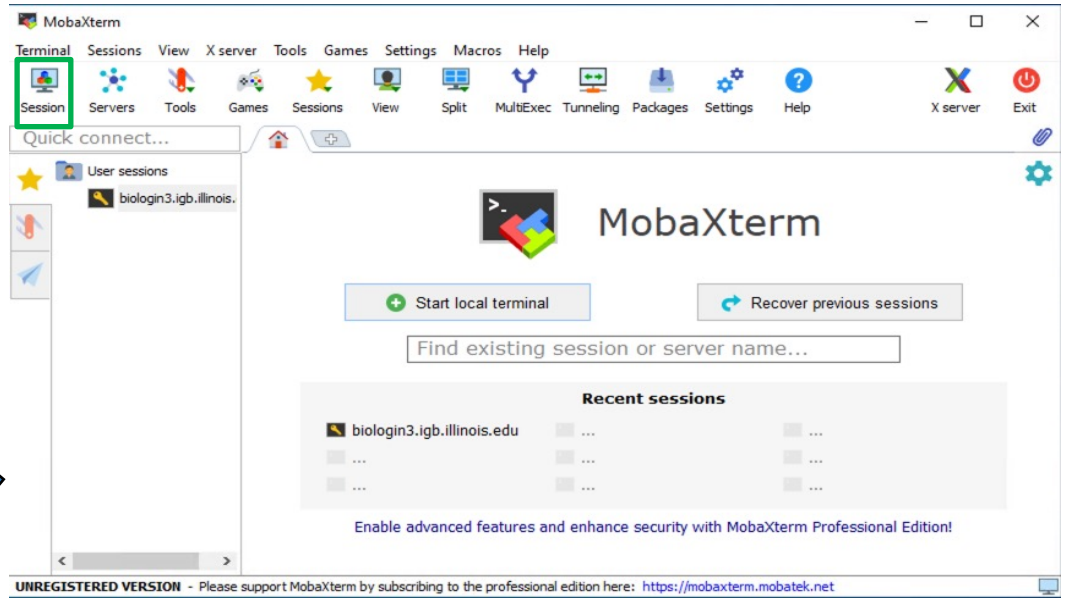
https://publish.illinois.edu/compgenomicscourse/2023-schedule/

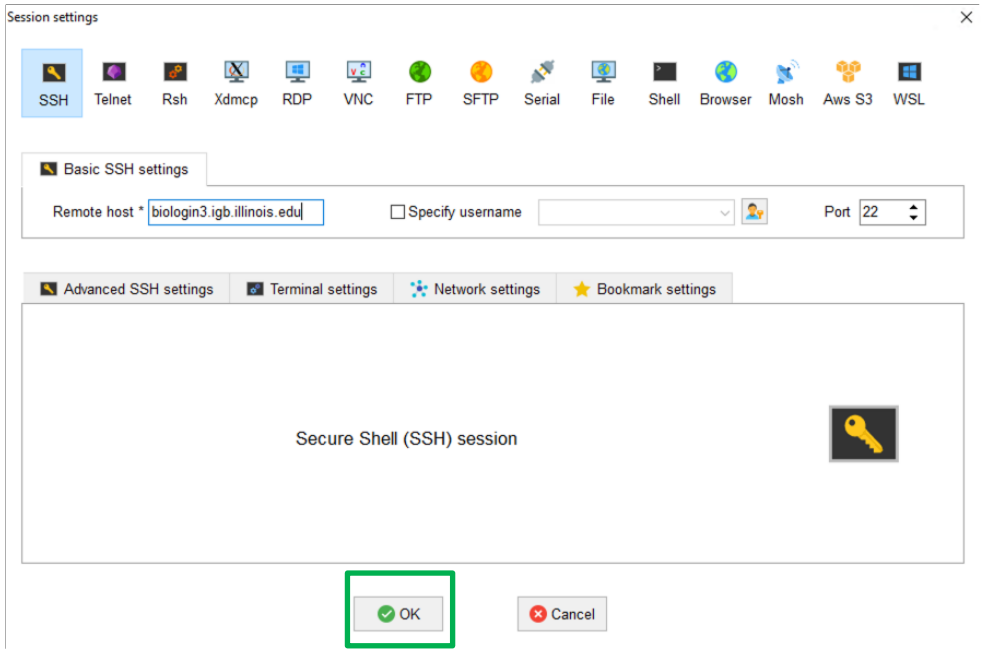
# Step 0A: Accessing the IGB Biocluster for First Time

- Open **MobaXterm** from the VM
- In a new session, select **SSH** and type the following host name:

  `biologin3.igb.illinois.edu`

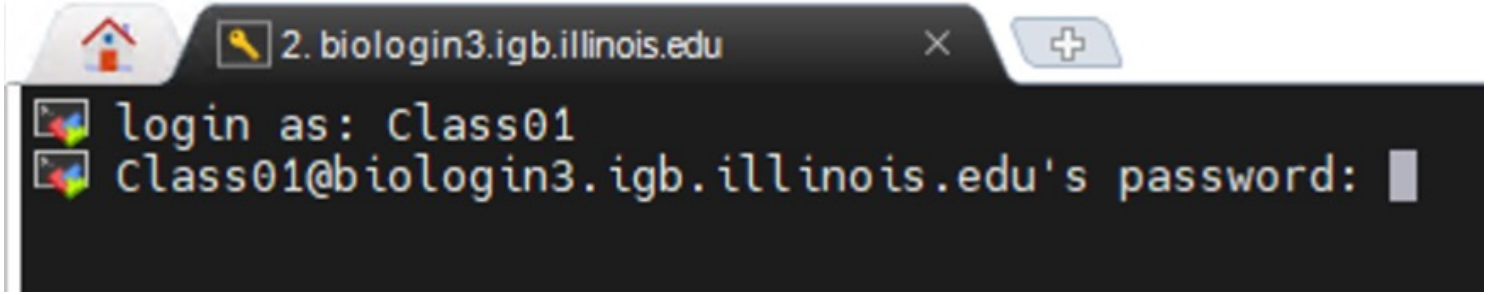
- Click **OK**



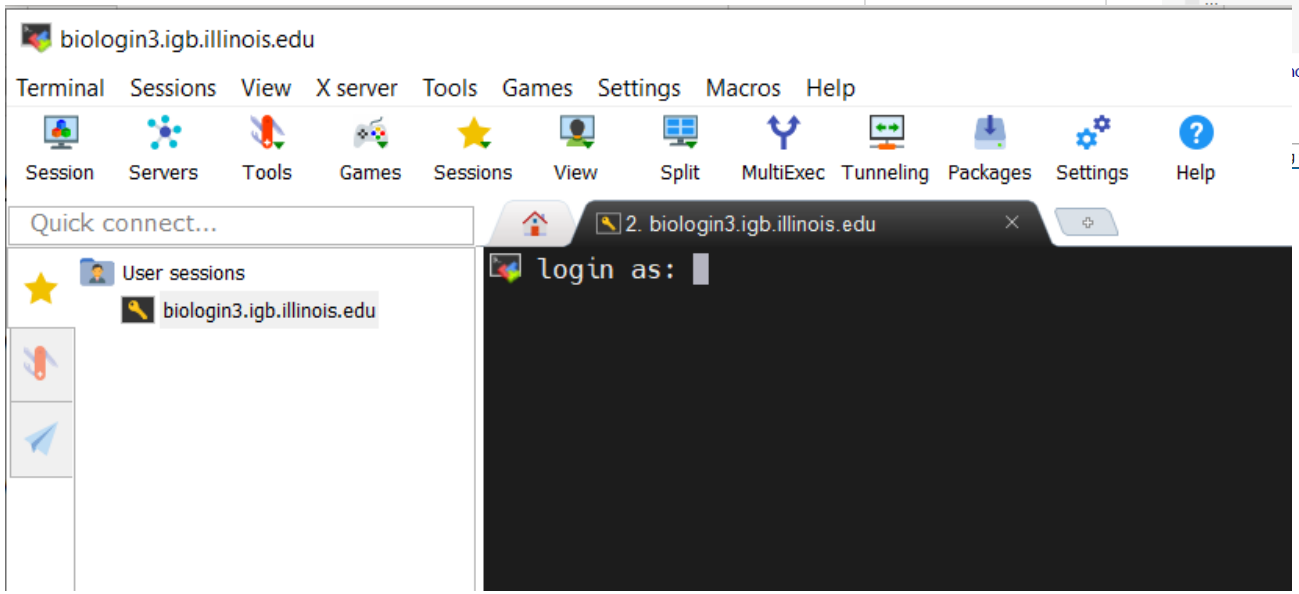
# Step 0A: Accessing the IGB Biocluster

- Enter login credentials assigned to you.
- Example username: C**lass01**
- You will not see any characters on screen when typing in password. Just type it.



# Step 0A: Accessing the IGB Biocluster

**If you have done this before**, just double-click on the session you created once and type username and password.



# Step 0B: Lab Setup

The lab is located in the following directory:

```
/home/classroom/mayo/2020/03_Variant_Calling/
```

This directory contains the data and results from the finished version of the lab (i.e. the version of the lab after the tutorial). Consult it if you are unsure about your runs. You don't have write permissions to the lab directory.

In the next slide, you will create a working directory of this lab in your home directory for your output to be stored. You will copy the necessary shell files (**.sh**) files from the data directory to your working directory.

Note **~** is a symbol in Unix paths referring to your home directory.

Note: In this lab, we will **NOT** login to a node on the biocluster. Instead, we will submit jobs to the biocluster.

# Step 0C: Lab Setup

Create a working directory called **~/03_Variant_Calling** in your home directory.

Copy all shell files (**.sh**) from the following path to your working directory.

**Copied Files**

`annotate_snpeff.sh`

`call_variants_ug.sh`

`hard_filtering.sh`

`post_annotate.sh`

```
$ mkdir ~/03_Variant_Calling

# Make working directory in your home directory

$ cd ~/03_Variant_Calling

# Change directory to your working directory.


$ cp /home/classroom/hpcbio/mayo_workshop/2019/Mayo-Variant-Calling/*.sh .

# Copy shell files to your working directory.
```

# Variant Calling Setup

In this exercise, we will use data from the 1000 Genomes project (EXOME, 60x coverage) to call variants, in particular single nucleotide polymorphisms.

The initial part of the GATK pipeline (**alignment, local realignment, base quality score recalibration**) has been done, and the BAM file has been reduced for a portion of human chromosome 20. This is the data we will be working with in this exercise.

# Step 1A: Running a Variant Calling Job

In this step, we will start a variant calling job using the **sbatch** command.

Additionally, we will gather statistics about our job using the **squeue** command.

```
$ sbatch call_variants_ug.sh

# This will execute call_variants_ug.sh on the biocluster.


$ squeue –u $USER

# Get statistics on your submitted job
```

# Step 1B: Output of Variant Calling Job

Periodically, call **squeue** to see if your job has finished.

You should have **4** files when it has completed.

**Files**

```
raw_indels.vcf

raw_indels.vcf.idx

raw_snps.vcf

raw_snps.vcf.idx
```

**Discussion**

- What did we just do?

We ran the GATK UnifiedGenotyper to call variants.

Look at the file structure.

# Step 1C: SNP and Indel Counting

In this step, we will count the # of SNPS and Indels identified in the **raw_snps.vcf** and **raw_indels.vcf** files.

We will use **grep**, which is a text matching program.

```
$ grep -c -v '^#' raw_snps.vcf                    # Get the number of SNPs.
# -v Tells grep to show all lines not beginning with # in raw_snps.vcf.
# -c Tells grep to return the total number of returned lines.
# Output should be approx. 14400.

$ grep -c -v '^#' raw_indels.vcf                  # Get the number of indels.
# Output should be approx. 1069.
```

# Step 1D: SNP and Indel Counting in dbSNP

In this step, we will count the number of SNPs and Indels in dbSNP.

dbSNP SNPs and Indels have the **rs#** identifier where # is a number.

**Example: rs1000**

```
$ grep -c 'rs[0-9]*' raw_snps.vcf       # Get the number of dbSNP SNPs.
# Return all lines in raw_snps.vcf containing rs followed by a number.
# -c Tells grep to return the total number of returned lines.
# Output should be approx. 12650.

$ grep -c 'rs[0-9]*' raw_indels.vcf   # Get the number of dbSNP indels.
# Output should be approx. 958.
```

# Step 2A: Hard Filtering Variant Calls

We need to filter these variant calls in some way.

In general, we would filter on quality scores. However, since we have a very small set of variants, we will use hard filtering.

```
$ sbatch hard_filtering.sh

# Execute hard_filtering.sh on the biocluster.

$ squeue -u $USER
```

Output Files

hard_filtered_snps.vcf

hard_filtered_indels.vcf

Periodically, call squeue to see if your job has finished.

# Step 2A: Hard Filtering Variant Calls

- Hard filtering steps use specific cutoffs with annotations for filtering data, labeling them according to any filters that a variant doesn't pass.
- In the lecture we discuss ways to look at samples and assess a cutoff
- Here we use example cutoffs recommended by GATK group

**Do not need to run this code**

```
gatk -T VariantFiltration \
    -R $REFERENCE \
    --variant $SNP_VCF_FILE \
    --clusterSize 3 \              # 3 SNPs within
    --clusterWindowSize 10 \       # 10nt window
    --filterExpression "QD < 2.0" \  # Condition
    --filterName "QDFilter" \        # Label
    --filterExpression "MQ < 40.0" \
    --filterName "MQFilter" \
    --filterExpression "FS > 60.0" \
    --filterName "FSFilter" \
    --filterExpression "HaplotypeScore > 13.0" \
    --filterName "HaplotypeScoreFilter" \
    --filterExpression "MQRankSum < -12.5" \
    --filterName "MQRankSumFilter" \
    --filterExpression "ReadPosRankSum < -8.0" \
    --filterName "ReadPosRankSumFilter" \
    -o $SNP_VCF_FILE_OUT
```

# Step 2B: Hard Filtering Variants Calls

In this step, we will count the # of filtered SNPs and Indels.

```
$ grep -c 'PASS' hard_filtered_snps.vcf   # Count # of passes

# Output 8554.


$ grep -c 'PASS' hard_filtered_indels.vcf  # Count # of PASSES

# Output 1069
```

## Discussion

1. Did we lose any variants?
2. How many *PASSED* the filter?
3. What is the difference in the filtered and raw input?
4. Why are these approximate (why do results slightly differ)? UnifiedGenotyper is non-deterministic when using multi-threading (timing)

# Step 2B: Hard Filtering Variants Calls

Some of the filters are as following:

- **QD:** variant confidence/ quality by depth, **QD < 2**
- **MQ:** RMS mapping quality, **MQ < 40**
- **FS:** Phred-scaled p-value, **FS > 60.0**

- What is the difference in the filtered and raw input?

In the filtered input the “FORMAT” column has information on whether the SNP has passed all the filters (“PASS”) or has failed any of them.

# Step 3A: Annotating Variants With SnpEff

With our filtered variants, we now need to annotate them with **SnpEff.**

**SnpEff** adds information about where variants are in relation to specific genes.

Periodically, call squeue to see if your job has finished.

```
$ sbatch annotate_snpeff.sh

# This will execute snpeff.sh on the biocluster.


$ squeue -u $USER
```

Output Files

hard_filtered_snps_annotated.vcf

hard_filtered_indels_annotated.vcf

# Step 3B: Annotating Variants With SnpEff

The IDs for the human assembly version we use are from Ensemble. The Ensemble format is ENSGXXXXXXXXXX.

Example: FOXA2's Ensemble ID is ENSG00000125798.

In this step, we would like to see if there are any variants of FOXA2.

```
$ grep -c 'ENSG00000125798' hard_filtered_snps_annotated.vcf

# Get the number of SNPS in FOXA2, ENSG00000125798.

# Output should be 3.


$ grep -c 'ENSG00000125798' hard_filtered_indels_annotated.vcf

# Get the number of Indels in FOXA2, ENSG00000125798.

# Output should be 0.
```

# Step 4: GATK Variant Annotator

**SnpEff** adds a lot of information to the VCF.

GATK Variant Annotator helps remove a lot of the extraneous information.

```
$ sbatch post_annotate.sh

# This will execute post_annotate.sh on the biocluster.


$ squeue -u $USER
```

Output Files

hard_filtered_snps_final.vcf

hard_filtered_indels_final.vcf

Exit MobaXterm by either closing the window or typing ‘exit’ in the command prompt.

# Visualization of Results

In this exercise, we will visualize the results of the previous exercise using the Integrated Genomics Viewer (IGV).

We are going to do visualization on VM.

# Step 0: Local Files

For viewing and manipulating the files needed for this laboratory exercise, the path on the VM will be denoted as the following:

**[course_directory]**

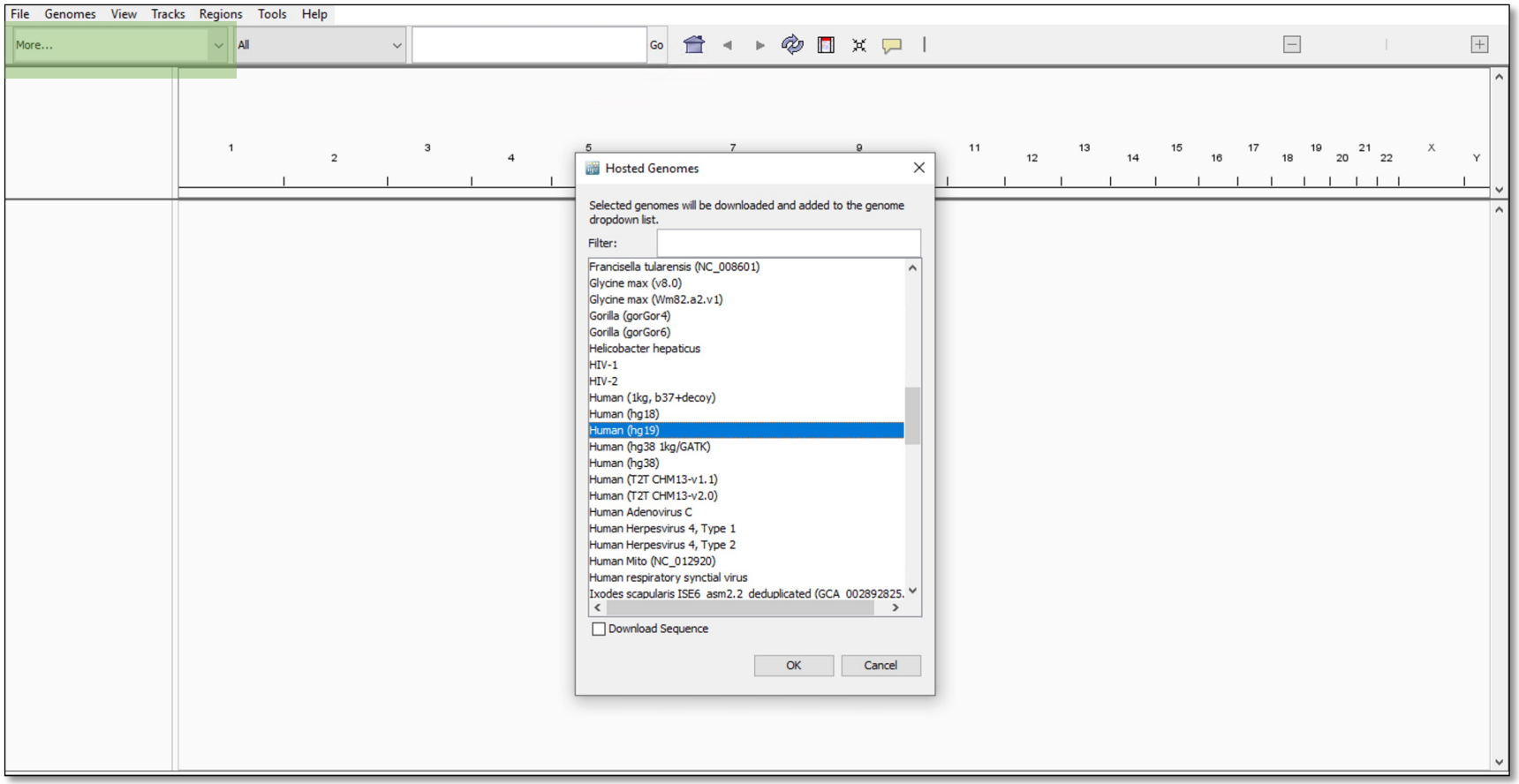
We will use the files found in:

**[course_directory]\03_Variant_Calling\results**

**[course_directory]= Desktop\VM**

# Step 5A: Visualization With IGV

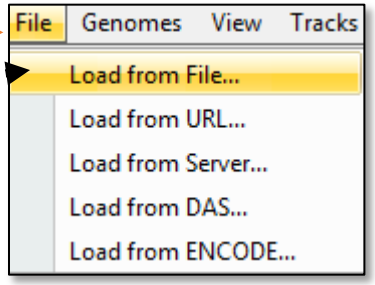
Switch the genome to **Human (hg19)**, which can be found by selecting **More...** under the genome selection panel



# Step 5B: Loading VCF Files

On the menu bar, click **File**

Click **Load from File…**

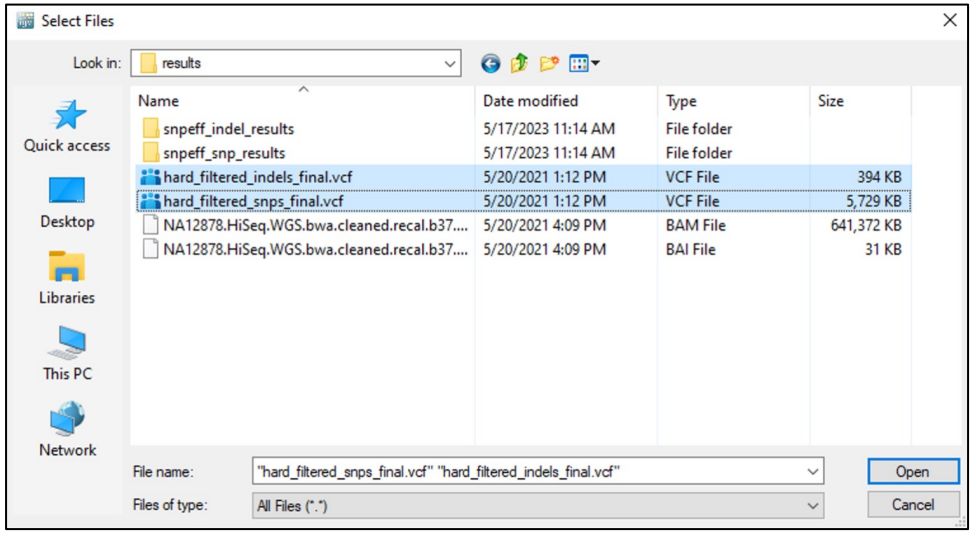



Navigate to: `[course_directory]/03_Variant_Calling/results`

Hold the **Ctrl** key down.

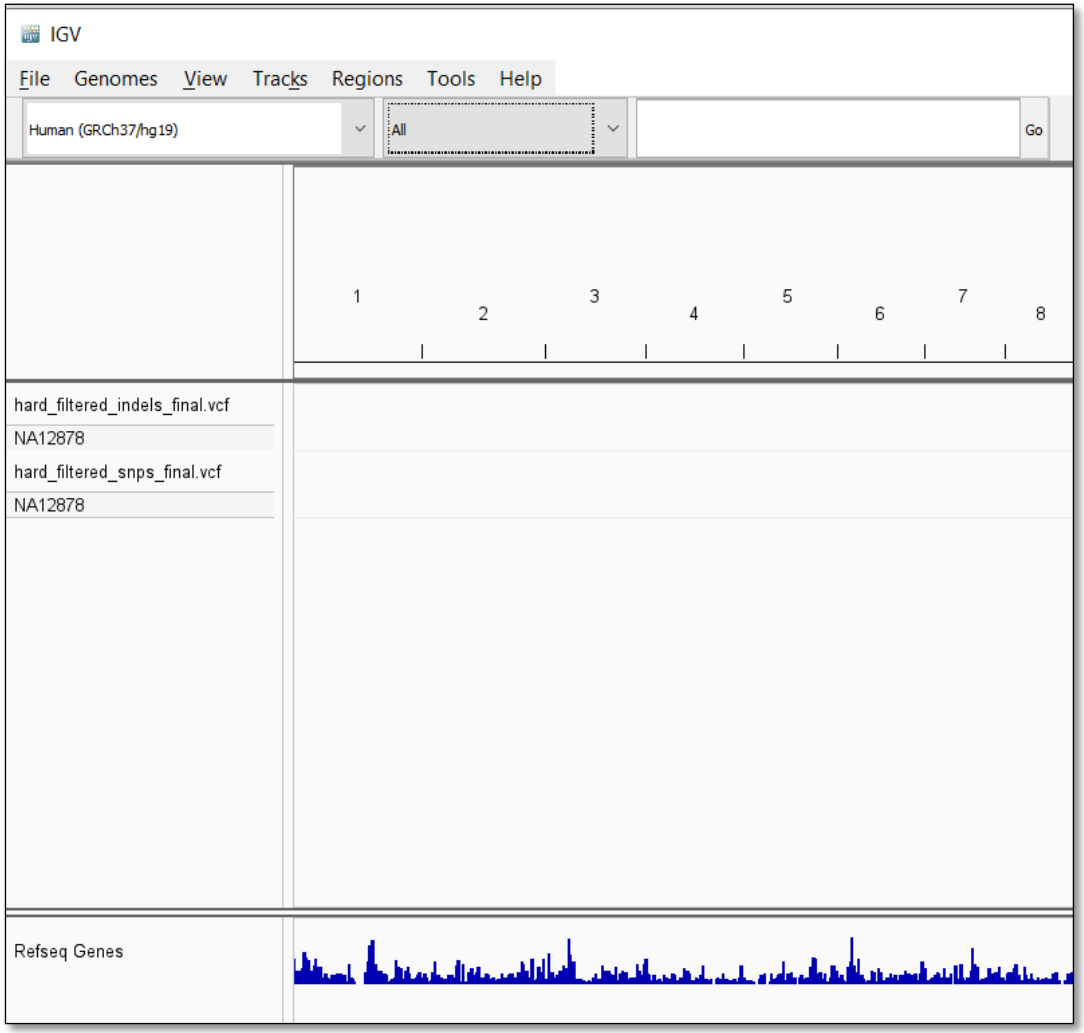
Click both **vcf** files.

Click **Open.**



# Step 5C: Loading VCF Files

You should see a windows similar to below:

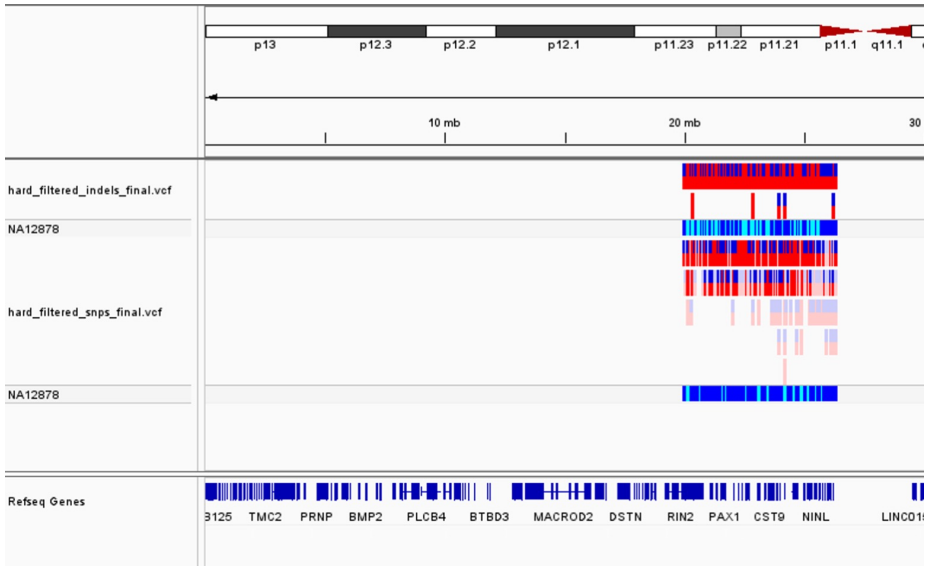


# Step 5D: Navigate to Chromosome 20

In the search box, type **chr20**.

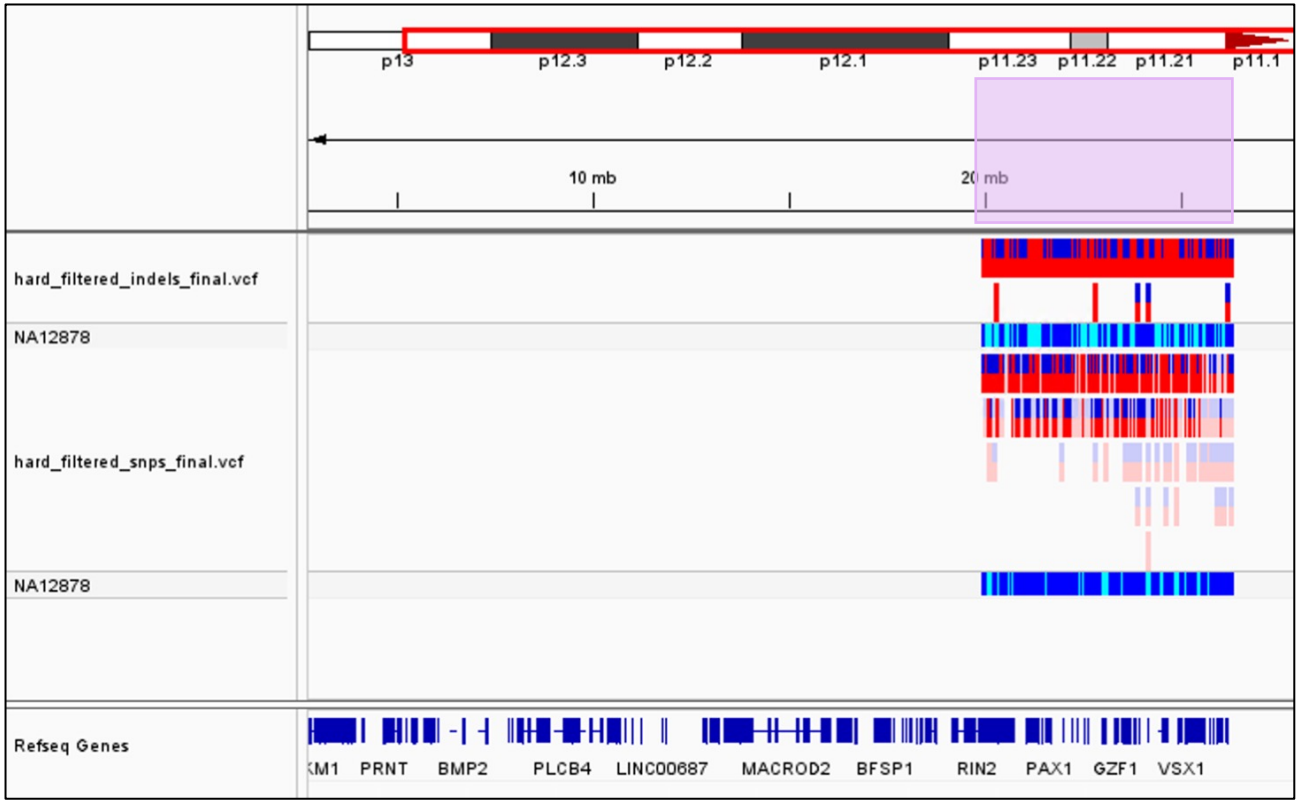

Press **Enter** or click **Go.**

You should see a track similar to the screenshot on the right.



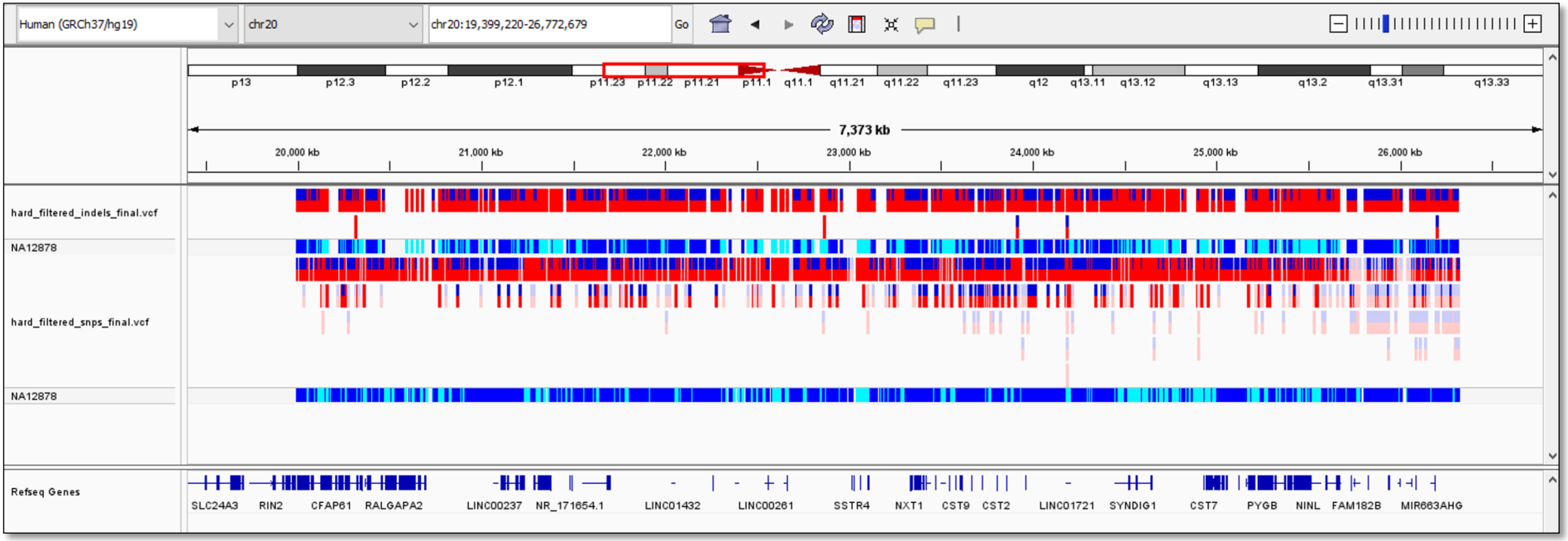
# Step 5E: Navigate to Chromosome 20

Click and drag from around the **20 mb** mark to about the **27 mb** mark.



# Step 5F: Navigate to Chromosome 20

The result should look similar to the screenshot below:



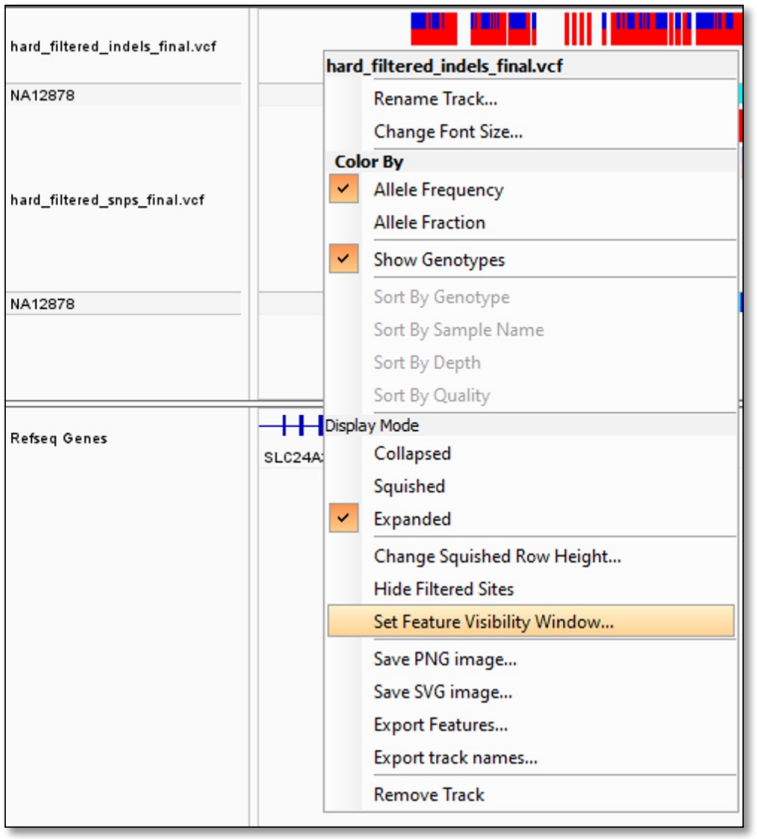
# Step 5G: Setting Feature Visibility Window

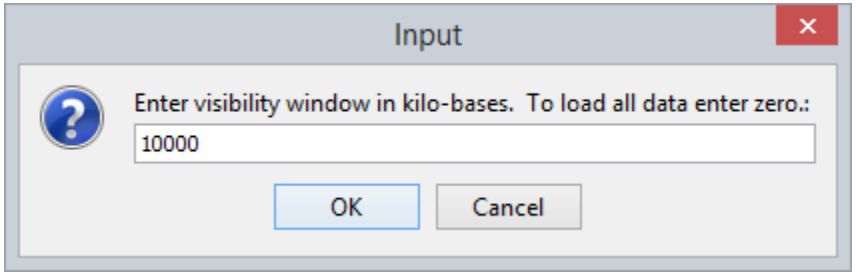
Do this for each VCF track:

**Right Click** and Select **Set Feature Visibility Window**

Enter **10000** (which is 10 Mb).

Click **OK.**

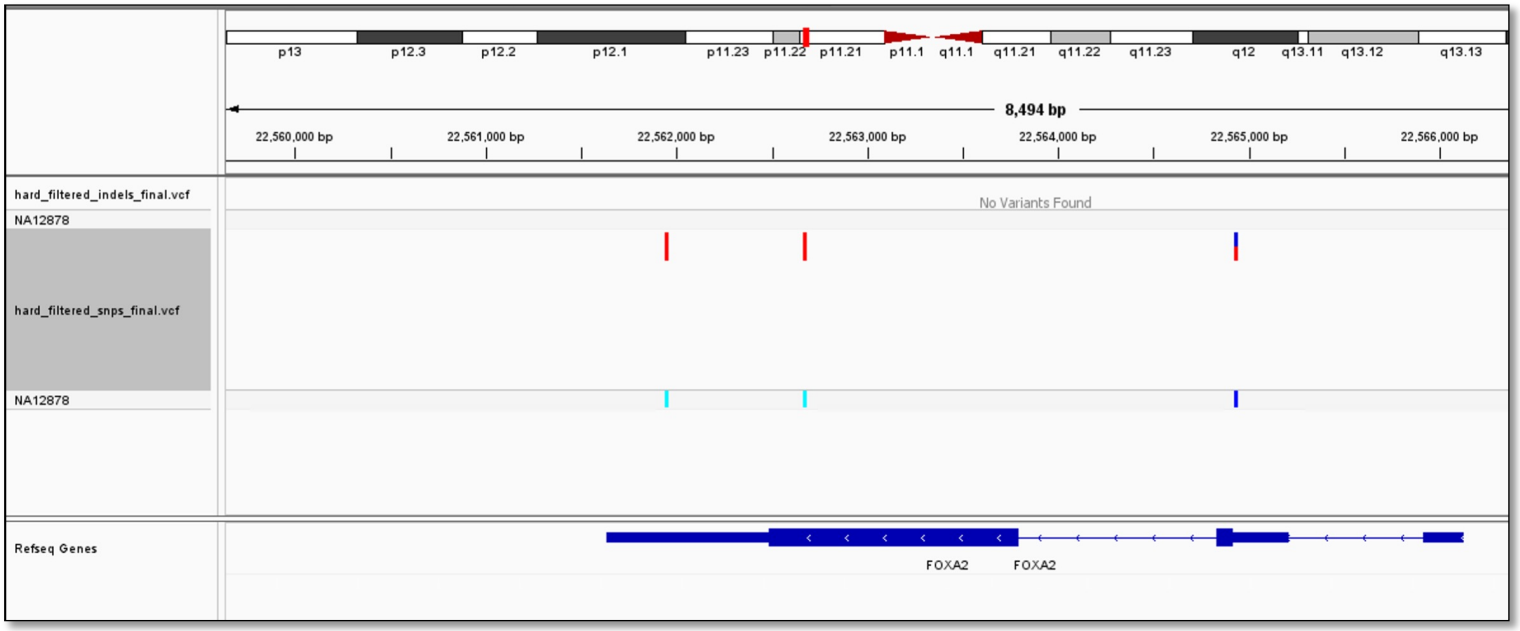


# Step 5H: Viewing FOXA2 Polymorphisms

In the search box, type **FOXA2** and press **Enter.**


You should see something like the window below:



# Checkpoint: FOXA2 Polymorphisms

1. How many SNPs are here?

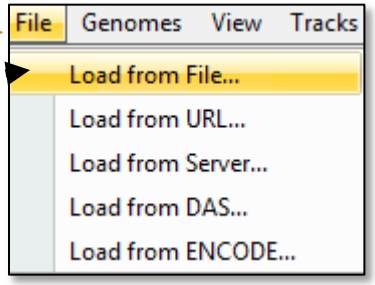
2. How many Indels are here?

3. How many SNPs are heterozygotes?

# Step 6A: Loading a BAM File

On the menu bar, click **File**

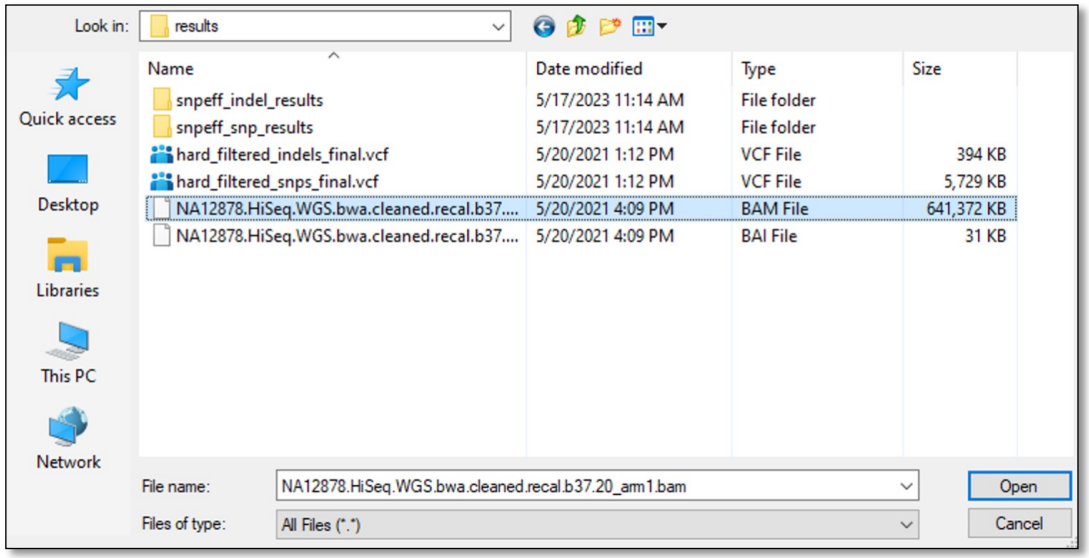
Click **Load from File…**




Navigate to: `[course_directory]/03_Variant_Calling/results`

Click the **bam** file.

Click **Open.**



# Step 6B: Loading BAM File

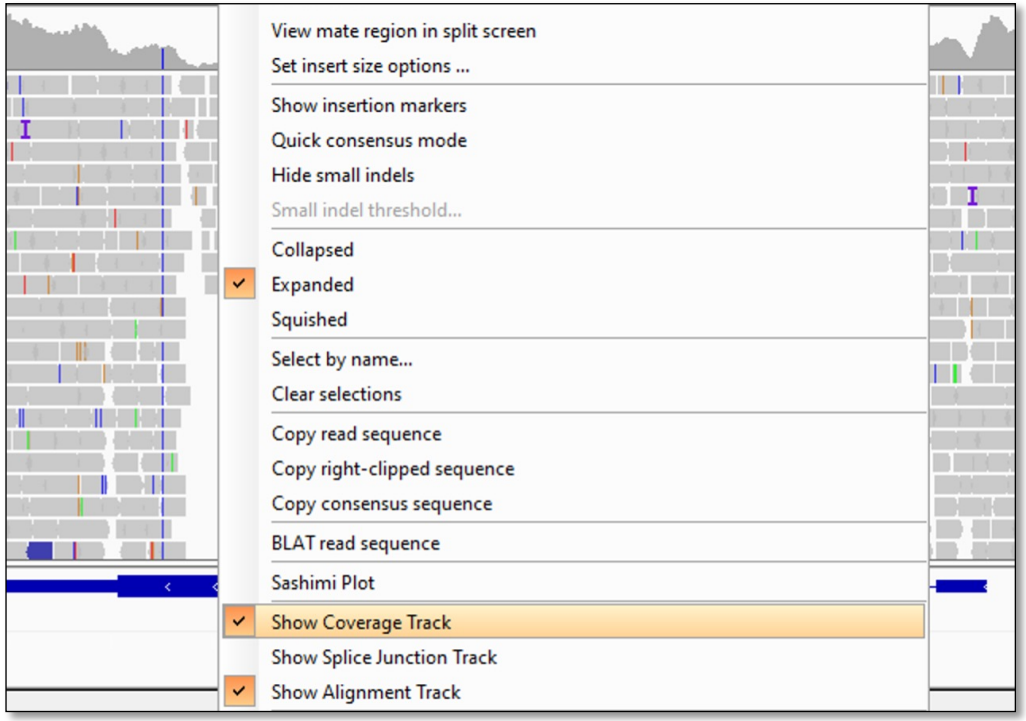
You should see a window with a new track similar to the one below:

# Step 6C: Show Coverage Track

Note: If you don't see a summary track like below :


Right **Click** on the **BAM** track.

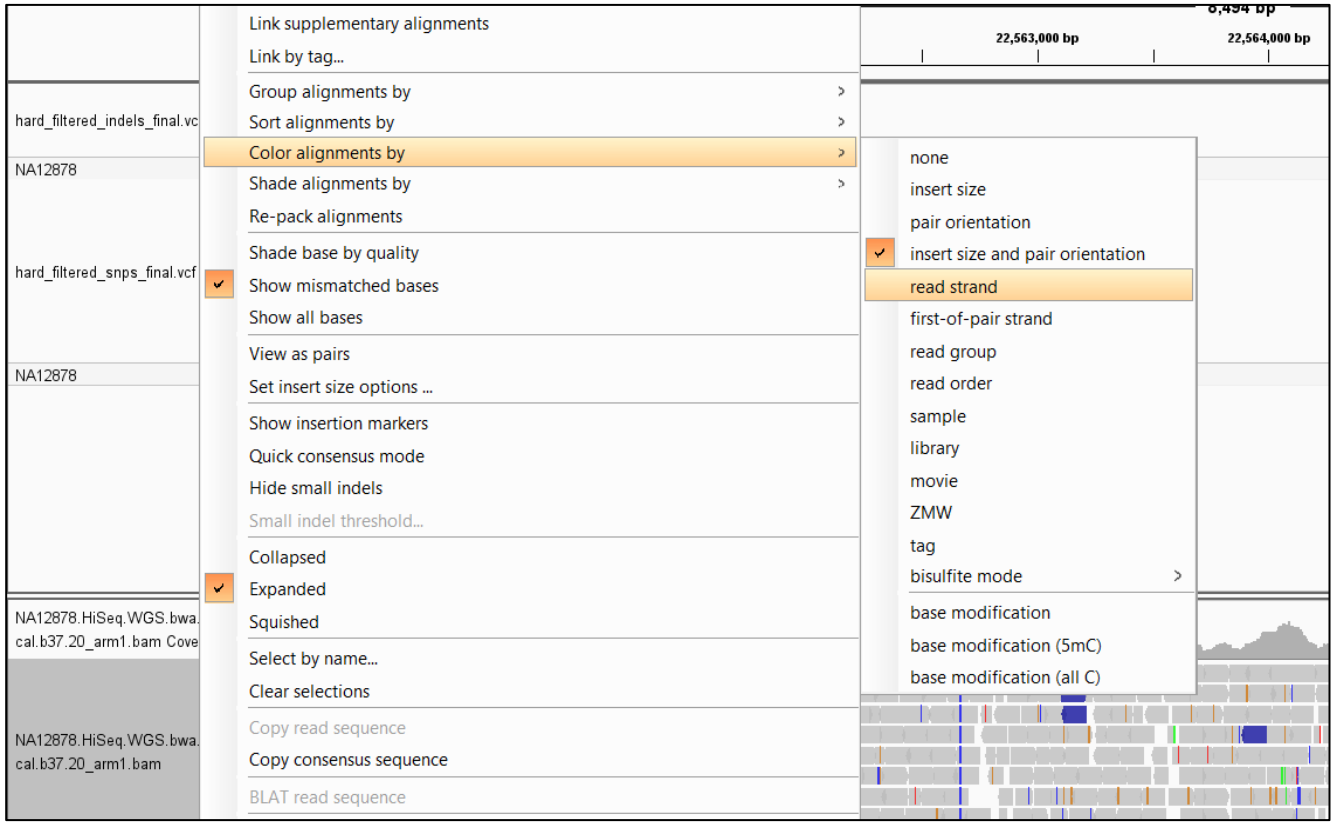
Click **Show Coverage Track**.



# Step 6D: Color Alignments by Read

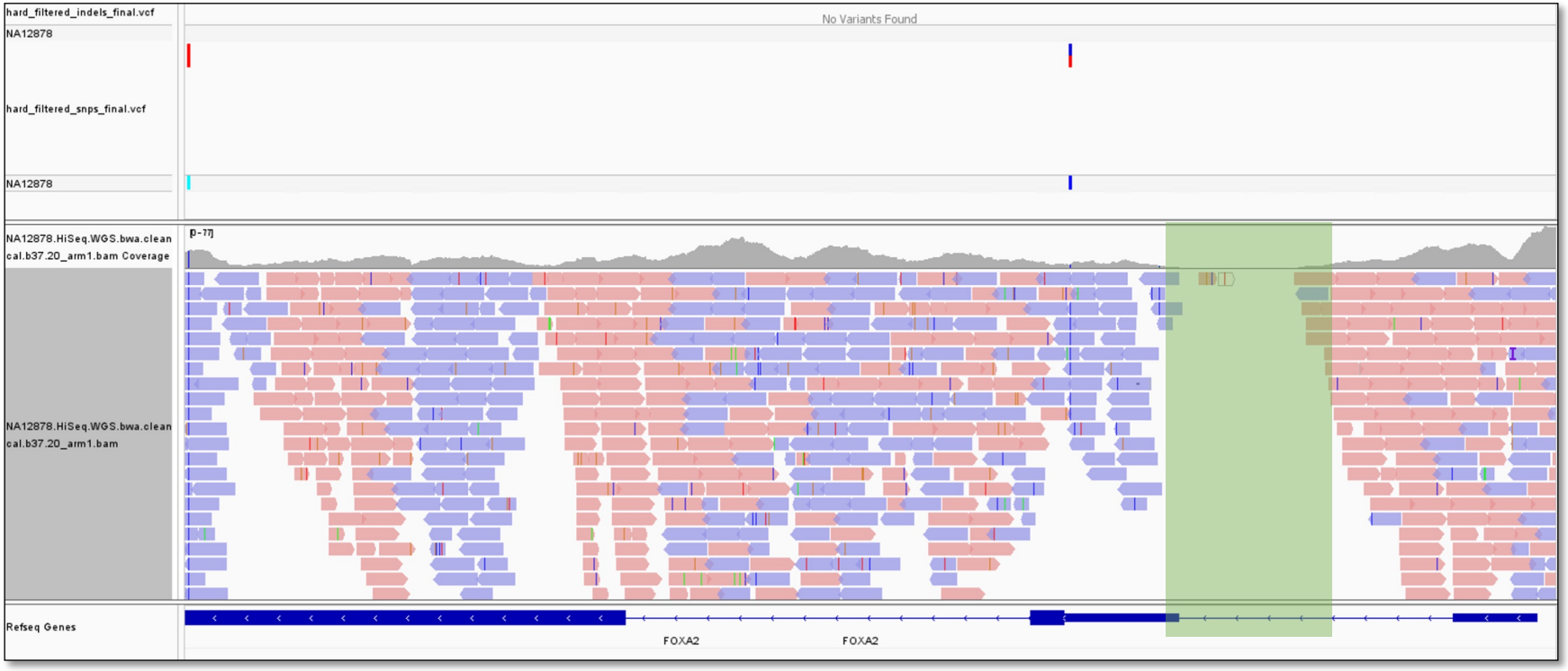
Right **Click** on the **BAM** track.

Click **Color Alignment by** and then **Read Strand**



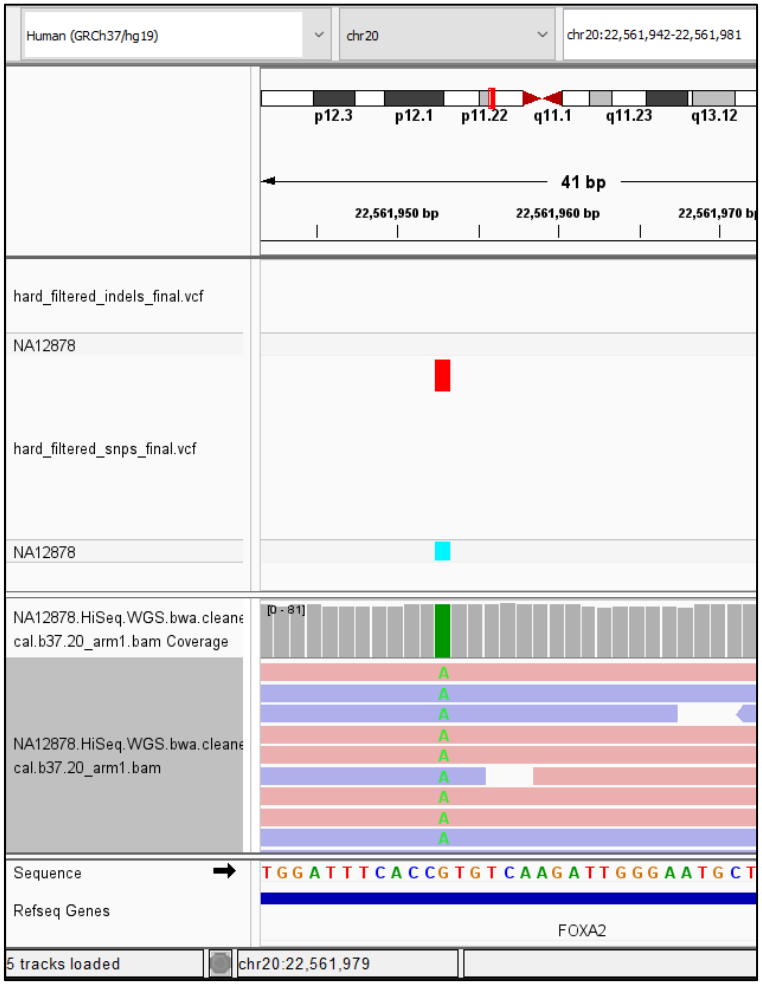
# Step 6E: FOXA2 Read GAP Question

What is happening in the highlighted portion?



# Step 6F: Viewing SNP Calls

Zoom In (double click) on SNPs to see the base pair calls on each read.



# Step 7: SnpEff Results

SnpEff gives a nice summary HTML file.

Navigate to the results directory for this lab:

```
[course_directory]\03_Variant_Calling\results
```

Open `snpEff_summary.html` in each of the following sub directories:

1. `snpeff_snp_results`
2. `snpeff_indel_results`

Browse each of the HTML files and note the results of the following slides:

# Step 7B: SNPEff Summary of SNPS

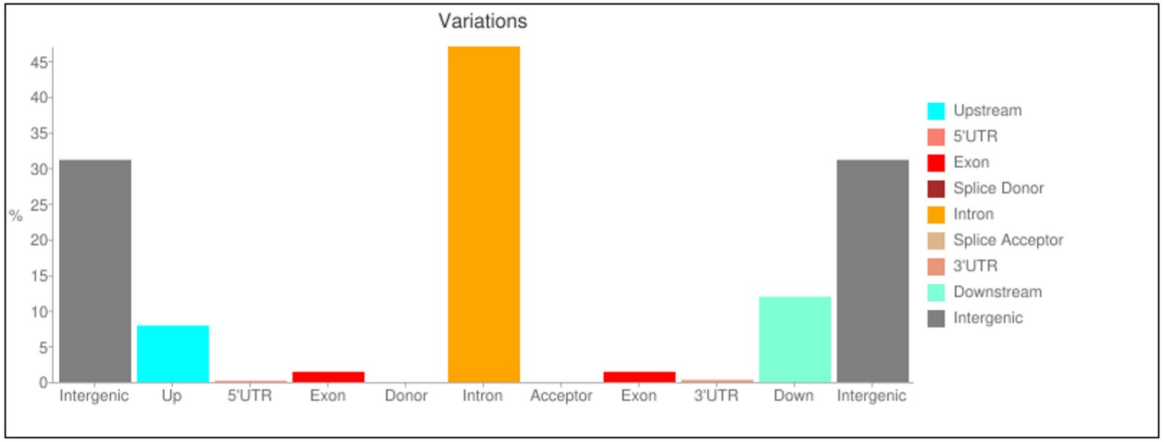
**Number of effects by type and region**

**Type**

| Type (alphabetical order) | Count | Percent |
|---|---|---|
| DOWNSTREAM | 3,890 | 11.972% |
| EXON | 331 | 1.019% |
| INTERGENIC | 10,119 | 31.143% |
| INTRON | 15,283 | 47.036% |
| NON_SYNONYMOUS_CODING | 76 | 0.234% |
| PROTEIN_STRUCTURAL_INTERACTION_LOCUS | 1 | 0.003% |
| SPLICE_SITE_ACCEPTOR | 2 | 0.006% |
| SPLICE_SITE_DONOR | 1 | 0.003% |
| START_GAINED | 6 | 0.018% |
| STOP_GAINED | 4 | 0.012% |
| SYNONYMOUS_CODING | 57 | 0.175% |
| UPSTREAM | 2,576 | 7.928% |
| UTR_3_PRIME | 101 | 0.311% |
| UTR_5_PRIME | 45 | 0.138% |

**Region**

| Type (alphabetical order) | Count | Percent |
|---|---|---|
| DOWNSTREAM | 3,890 | 11.973% |
| EXON | 469 | 1.444% |
| INTERGENIC | 10,119 | 31.146% |
| INTRON | 15,280 | 47.031% |
| SPLICE_SITE_ACCEPTOR | 2 | 0.006% |
| SPLICE_SITE_DONOR | 1 | 0.003% |
| UPSTREAM | 2,576 | 7.929% |
| UTR_3_PRIME | 101 | 0.311% |
| UTR_5_PRIME | 51 | 0.157% |



# Step 7C: SNPEff Summary of Indel Lengths

The summary of **snpeff indels** shows the following distribution of indel lengths: