Variant Calling Workshop

Chris Fields

PowerPoint by Casey Hanson
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.
Step 0A: Accessing the IGB Biocluster

Open **Putty.exe**

In the **hostname** textbox type:

```
biologin-0.igb.illinois.edu
```

Click **Open**

If popup appears, Click **Yes**

Enter login credentials assigned to you; example, user **class00**.

Now you are all set!
Step 0B: Lab Setup

The lab is located in the following directory:

```
/home/classroom/mayo/2019/03_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 unsure about your runs. You don’t have write permissions to the lab directory.

Create a working directory of this lab in your home directory for your output to be stored. Note `~` is a symbol in Unix paths referring to your home directory. Copy the necessary shell files (.sh) files from the data directory to your working 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: Local Files

For viewing and manipulating the files needed for this laboratory exercise, insert your flash drive.

Denote the path to the flash drive as the following:

```
[course_directory]
```

We will use the files found in:

```
[course_directory]/03_Variant_Calling/results
```
Create a working directory called `~/03_Variant_Calling` in your home directory.

Copy all shell files (.sh) from the data directory to your working directory.

```
$ mkdir ~/03_Variant_Calling

# Make working directory in your home directory

$ cp /home/classroom/mayo/2019/03_Variant_Calling/data/*.sh ~/03_Variant_Calling

# Copy shell files to your working directory.
```
Variant Calling Setup

In this exercise, we will use data from the 1000 Genomes project (WGS, 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.

```bash
$ cd ~/03_Variant_Calling

# Change directory to your working directory.

$ sbatch call_variants_ug.disable.sh

# This will execute call_variants_ug.disable.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.

Discussion

What did we just do?

We ran the **GATK UnifiedGenotyper** to call variants.

Look at file structure.

<table>
<thead>
<tr>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw_indels.sh</td>
</tr>
<tr>
<td>raw_indels.vcf.idx</td>
</tr>
<tr>
<td>raw_snps.vcf</td>
</tr>
<tr>
<td>raw_snps.vcf.idx</td>
</tr>
</tbody>
</table>
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 the program 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**

```bash
$ 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 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)?
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 `ENSGxxxxxxxxxxxxx`.

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
```
Visualization of Results

In this exercise, we will visualize the results of the previous exercise using the Integrated Genomics Viewer (IGV).
Step 5A: Visualization With IGV

Switch the genome to **Human (b37)**.
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`

Hold the **Ctrl** key down.

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 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
Step 7B: SNPEff Summary of Indel Lengths

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

![Insertion deletion length histogram](image)