Syed Shahzaib Ali

Project 3: Bcf Tools and Freebays Variant Caller Comparison

March 19th, 2021

Introduction

Deep genome analysis is an important practice for identifying accurately diagnostic, prognostic, and predictive determinants (Bohannan and Mitrofanova 2019). One of these types of analysis is variant calling an important statistical and computational procedure that discriminates between true mutations errors and experimental errors (Bohannan and Mitrofanova 2019).

The types of variants that are identified are for example single nucleotide polymorphisms (SNP's), DNA insertions and deletions (Indels) etc. The popular softwares that are used to identify these variants are GATK, samtools, Freebays and TVC. In this project, Freebays and bcftools were used for variant calling and recognizing the SNP's. The Sam tools package contains two parts: Sam tools and bcftools. The bcftools is noted to be used for variant calling and has been recognized to be quite efficient for SNP variant calling on Illumina data according to Hwang et al 2015. However, a previous study by Li 2011 showed that that bcftools works much better with a variant call format (VCF) rather than a binary variant call format (BCF). It has also been determined that bcftools (samtools) is biased towards adding the reference allele (AR) which then causes a high probability of AR errors than IR (ignore reference allele) errors (Hwang et al 2015). Thus, for heterozygous SNP calling bcftools caution is suggested. However, considering the Hwang et al study bcftools is very versatile and known to perform well with both ion proton datasets and Illumina datasets. This means it carries a lower risk as compared to other popular ones for variant calling especially in the case of SNP determination.

In-terms of Freebays as a variant caller it has been noted to be biased to ignore the reference allele which than causes it to ignore the heterozygous SNP calls and report mostly homozygous SNP. Unlike Samtools, Freebays has been determined to only be a good performer with Illumina data meaning it is where the least errors are made, while with ion proton data it produces a lot. This is where the difference between bcftools and Freebays significant as BCF tools is very versatile according to the information by Hwang et al 2015. In additions, Freebays versatility comes when considering the aligning software. It performs well with any, while bcftools works the best with bwa mem software.

Considering the information presented by Hwang et al 2015 this paper has explored Freebays and bcftools for SNP variant calls to see the major differences. The data used are 10 burbot genome aligned files which were aligned with the help of bwa software. Considering Hwang et al it is determined that bcftools will perform better than Freebays.

Results

The single nucleotide polymorphisms determined by the BCF variant calling method and Freebays variant calling method were 151,497 and 160,695. After the filtering was done which was conducted through VCF tools with a missing value score threshold of greater than 20 the SNPs values were reported to be 61,846 and 106,117. The minor allele frequency distribution was very uneven with most values being reported 0, and this was observed for both allele frequencies obtained through variant files of both different softwares. The number of SNPs where the allele frequency was 0.5 was much higher in the Freebays variant file. The frequencies histogram is shown in figure1. The depth distribution was much less as shown in figure 1 for BCF when compared to Freebays. The SNPs that overlapped between the two softwares VCF files were 39,933. The SNPs determined by bcftools that overlapped onto Freebays determined SNPs were 21,913 and the SNPs determined by Freebays tools that overlapped onto bcftools determined SNPs were 66,184.

Discussion

Type of Variant Call	SNP (Pre-Filter)	SNP (Filtered)	SNP (Overlapped Between)	SNP (Overlap Individual)	Percentage Overlap
BCF tools	151497	61846	39933	21913	35.4%
Freebays	160695	106117	39933	66184	62.3%

As shown in figure 1 the minor allele frequencies plots show that the chromosomes present in the burbot samples are a large number homozygous as the minor allele frequency values were 0. This is something to be true and was a conclusion reach by both the Freebays and bcftools software. Freebays however, captured more heterozygous SNPs than bcftools as a large number was 0.5 in comparison to bcftools as shown in the plots. Given the SNP depth plots shown in figure 1, it appears that there seems to be an abundance of low coverage SNP sites as most depths were less than 10. This was the result using either the bcftools or Freebays. Freebays on the hand did provide a large number of SNPs that had much better depth or much better coverage which means that it was able to perform much better than bcftools.

In addition to the plots, the VCF files generated were to have a quality score greater than 20 and given that the Freebays tool generated a large number of sites (106,117) in comparison to bcftools, it appears that Freebays has indeed performed much better than the bcftools. Given the conclusion from the results, it seemed to disagree with Hwang et al 2015 that samtools was a much better performer in their research in comparison to Freebays especially given that bwa mem software and illumine data was involved. The reason for this disagreement between the results could be due to the different type of filter involved as well as the type of data they were working with. So, in the future perhaps different commands in the VCF tools filter could be an option as the type of filter you will use could make a large amount of difference.

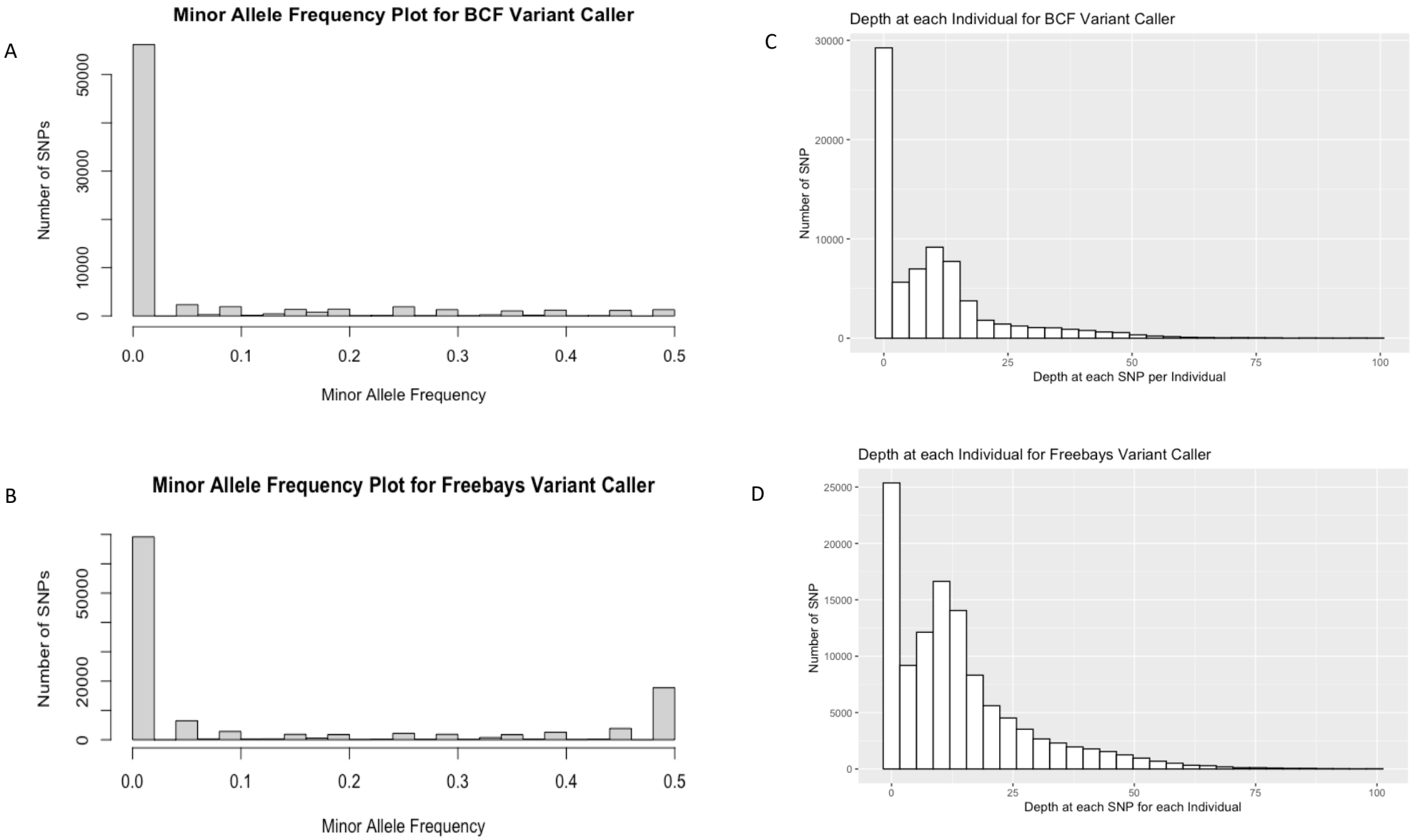

Figure 1 shows the plots for minor allele frequency and SNP depth for each loci. 1a) minor allele frequency when BCF was used as a variant caller for burbot aligned files obtained through use of BWA software. 1b) minor allele frequency when freebays was used as a variant caller for burbot aligned files obtained through use of BWA software. 1c) sum SNP depth when BCF was used as a variant caller for burbot aligned files obtained through use of BWA software. 1d) sum SNP depth frequency when freebays was used as a variant caller for burbot aligned files obtained through use of BWA software.

Methods

BCF Variant Calling Method

There were two methods chosen to use as variant calls: BCF tools and Freebays. First bcftools was used to make VCF file. However, this was not executed in the command line but in batch submission script.

```
bcftools mpileup -f burbot_reference/GCA_900302385.1_ASM90030238v1_genomic.fna bwa_assem/*.sorted.bam | bcftools call -m --variants-only >10ind_bcftools.vcf
```

Once the variant file was made it was filtered where the geneome quality determined by missing value QUAL is greater than or equal to 20 and outputted to a new variant folder.

```
bcftools filter -e '%QUAL>=20' 10ind_bcftools.vcf > final.vcf
```

Once it was filtered the SNP values were determined by the BCF tools command mentioned below.

```
bcftools stats 10ind_bcftools.vcf| grep "number of SNPs"
```

```
bcftools stats final.vcf| grep "number of SNPs"
```

After determining the numbers of SNPs in each file the minor allele frequency was determined as well as the depth for each SNP in the Unix system and R.

```
grep -oP 'DP=\d+' final.vcf > depth_perlocus.txt
```

```
sed 's/DP=//g' depth_perlocus.txt > depth_number_bcf.txt
```

The package in R that calculates the minor allele frequcecy is vcfR. First the VCF file was uploaded after downloading it on the local computer and then the frequency was calculated using maf() function after which the hist() function from base R was used to make the histogram for minor allele frequency for each SNP at each loci.

```
library(vcfR)
```

```
vcf = read.vcfR(file.choose())
```

```
maf = maf(vcf)
```

```
maf = as.data.frame(maf)
```

```
hist(maf$Frequency, main = "Minor Allele Frequency Plot for BCF Variant Caller", xlab = "Minor Allele Frequency", ylab = "Number of SNPs")
```

Finally the depth file was exported to local computer and a histogram was built using ggplot for each individual file for the 10 genome files.

```
library(ggplot)

depth.bcf = read.table(file.choose())

depth.bcf = depth.bcf %>% mutate(depth.ind = depth.bcf$V1/10)

depth.bcf %>% filter(depth.ind <100) %>% ggplot(aes(x=depth.ind)) +
geom_histogram(colour="black", fill="white") + xlab("Depth at each SNP per Individual") +
ylab("Number of SNP") + labs(title = "Depth at each Individual for BCF Variant Caller")
```

Freebays Variant Calling Method

First the readgroups were made for all the .sorted.bam files. This was done through a submission script.

```
module load StdEnv/2020 samtools/1.11 picard/2.23.3

bamfile = $1

basename = `echo $bamfile | sed 's/\.sorted.*\.bam.*//'`

echo "All Done"

java -jar $EBROOTPICARD/picard.jar AddOrReplaceReadGroups I=$bamfile
O=$basename.rg.sorted.bam RGID=$basename RGLB=lib1 RGPL=illumina RGPU=unit1
RGSM=$basename
```

Once the readgroup files were made samtools was used to index the files using the code below. The command execution remained the same, but the files were changed one by one. Due to some errors it had to be done one by one.

```
Samtools index 871_71_TRL_421.rg.sorted.bam
```

The freebays software was excuted to generate a variant file.

```
freebayes -f GCA_900302385.1_ASM90030238v1_genomic.fna -b
871_70_TRL_363.rg.sorted.bam -b 871_73_TRL_571.rg.sorted.bam -b
871_76_TOL_358.rg.sorted.bam -b 871_79_TOL_355.rg.sorted.bam -b
871_71_TRL_421.rg.sorted.bam -b 871_74_TRL_599.rg.sorted.bam -b
```

871_77_TOL_360.rg.sorted.bam -b 871_72_TRL_571.rg.sorted.bam -b 871_75_TRL_449.rg.sorted.bam -b 871_78_TOL_325.rg.sorted.bam > final.free.vcf
Then the VCF filter command was used and the depth for the SNPs at each loci was calculated.

vcffilter -f "QUAL > 20" 10ind_bcftools.vcf > bcf.new.vcf

vcftools --vcf freebays.new.vcf --site-depth -c > locusdepth2.txt

Finally using the same commands as mentioned for R as well as the same packages the minor allele frequency was calculated, and histograms were created. Furthermore, the SNPs that overlapped were determined between each variant callers. This was done after converting the files to a compressed version (.gz).

bcftools isec -p isec burbot_bcf_filter.vcf.gz burbot_free.vcf.gz