BINF 6110 Assignment 1:

Introduction:

This project was comprised of aligning sequence reads to a three-spine stickleback reference genome, to help identify the structural changes, variants and population-level differentiation. The reference genome was used to identify and discover SNPS of the population. The goal of this project was after alignment, the allelic variation and population differentiation were to be measured with different filtering parameters. This was done in effort to assess the genetic variation within the population and whether or not different parameters can lead to different results.

During the alignment process Sam tool was used to align the sequence read to reference genome. The loading of the Sam tool was initially a challenge as different particular versions of the tool needs to be loaded (samtools/1.20). However, once the tool was loaded, the SAM/BAM files were created and the alignment worked. Since the dataset was quite large, much higher processing power and memory was required, which took a long time for processes to complete. When measuring allelic variation the depth coverage parameters used were (DP>10 & DP>20), to remove low-confidence variants. The advantage of this parameter was it could reduce the false-positive variants present in low-depth regions, essentially impacting population diversity estimation. The disadvantage was that it may have excluded a rare variant in low-depth region. These parameters were chosen because low-depth variants can be unreliable since they can be sequencing errors rather than variants, therefore the threshold DP>20 was used to obtain high confidence variants, to achieve an accurate population diversity estimation.

Methods:

Path to directories: (chmod 755 /scratch/ebad/stickleback_project/ -R)

Codes Section:

(Description of code is above the code)

- **Loads SAMtools, Stacks, and BWA modules for sequence alignment, processing, and variant calling.**

module load samtools/1.20

module load stacks/2.67

module load bwa/0.7.17

- **Navigates to the project directory.**
- **Lists available sequence files.**
- **Copies FASTQ files from a shared directory to the working directory.**

cd /scratch/ebad/stickleback_project

ls /home/lukens/scratch/Assignment_1

cp /home/lukens/scratch/Assignment_1/*.fq.gz /scratch/ebad/stickleback_project/

- **Downloads the stickleback reference genome from Ensembl.**
- **Decompresses the genome file (if needed).**
- **Indexes the genome for alignment using BWA.**

wget https://ftp.ensembl.org/pub/release-113/fasta/gasterosteus_aculeatus/dna/Gasterosteus_aculeatus.GCAuleatus_UGA_version5.dna.toplevel.fa.gz

gzip: Gasterosteus_aculeatus.GCAuleatus_UGA_version5.dna.toplevel.fa.gz

bwa index /home/ebad/scratch/stickleback_project/Gasterosteus_aculeatus.GCAuleatus_UGA_version5.dna.toplevel.fa

- **Aligns each FASTQ file to the reference genome using BWA MEM with 4 threads.**
- **Outputs SAM files (Sequence Alignment Map).**

```
for sample in $(ls *.fq.gz | sed 's/.fq.gz//')
do
    bwa mem -t 4 Gasterosteus_aculeatus.GCAuleatus_UGA_version5.dna.toplevel.fa ${sample}.fq.gz > ${sample}.sam
done
```

- **Converts SAM files to BAM format (binary, more efficient storage).**

```
for sample in $(ls *.sam | sed 's/.sam//')
do
    samtools view -Sb ${sample}.sam > ${sample}.bam
done
```

- **Sorts the BAM files by genomic coordinate to improve efficiency.**

```
for sample in $(ls *.bam | sed 's/.bam//')
do
   samtools sort ${sample}.bam -o ${sample}_sorted.bam
done
```

- **Creates index files (.bai) for faster random access during variant calling.**

```
for sample in $(ls *_sorted.bam | sed 's/_sorted.bam//')
do
   samtools index ${sample}_sorted.bam
done
```

- **Uses mpileup to generate an intermediate variant dataset.**
- **Calls variants (SNPs and indels) and stores them in VCF format.**

```
bcftools mpileup -Ou -f Gasterosteus_aculeatus.GAculeatus_UGA_version5.dna.toplevel.fa
*_sorted.bam | \
bcftools call -mv -Oz -o stickleback_variants.vcf.gz
bcftools index stickleback_variants.vcf.gz
```

- **Filters out low-quality variants:**
- **QUAL < 20 (low confidence calls).**
- **DP < 10 (low read depth coverage).**
- **Saves the filtered VCF file.**

bcftools filter -Oz -o stickleback_variants_filtered.vcf.gz -s LOWQUAL -e 'QUAL<20 || DP<10' stickleback_variants.vcf.gz

- **Extracts SNP summary information (chromosome, position, reference/alternative alleles, quality).**
- **Counts the total number of SNPs in the filtered dataset.**
- **Saves variant quality scores for distribution analysis**.

bcftools query -f '%CHROM\t%POS\t%REF\t%ALT\t%QUAL\n' stickleback_variants_filtered.vcf.gz > snps_summary.txt

bcftools view -H stickleback_variants_filtered.vcf.gz | wc -l

bcftools view -v snps stickleback_variants_filtered.vcf.gz | wc -l

bcftools query -f '%QUAL\n' stickleback_variants_filtered.vcf.gz | sort -n | uniq -c > qual_distribution.txt

- **Filters SNPs based on depth (DP):**
- **MinDP10: Retains SNPs with at least 10 supporting reads.**
- **MinDP20: Applies a stricter threshold, keeping only SNPs with ≥20 reads.**
- **Indexes the new filtered VCF files.**

Default threshold

bcftools filter -e 'DP<10' -Oz -o stickleback_variants_minDP10.vcf.gz stickleback_variants.vcf.gz

bcftools index stickleback_variants_minDP10.vcf.gz

Higher threshold

bcftools filter -e 'DP<20' -Oz -o stickleback_variants_minDP20.vcf.gz

stickleback_variants.vcf.gz

bcftools index stickleback_variants_minDP20.vcf.gz

- **Computes variant statistics (e.g., transition/transversion ratios, depth distribution, allele frequencies) for different depth filters.**
- **Saves results in stats_minDP10.txt and stats_minDP20.txt.**

For DP >= 10

bcftools stats stickleback_variants_minDP10.vcf.gz > stats_minDP10.txt

For DP >= 20

bcftools stats stickleback_variants_minDP20.vcf.gz > stats_minDP20.txt

- **Gstacks processes sorted BAM files to:**
- **Identify loci.**
- **Call SNPs and genotypes.**
- **Uses a population map (`population_map.txt`) to assign individuals to populations.**
- **Outputs results to `gstacks_output/`.**

gstacks -I ./ -M population_map.txt -S '.sorted.bam' -O gstacks_output

- **Runs the populations program in Stacks to:**
- **Compute F-statistics (population differentiation).**
- **Output VCF, STRUCTURE, and single-SNP files.**
- **Uses 4 CPU threads for faster execution.**

- **Saves results in `output_directory/`.**

populations -P gstacks_output -M population_map.txt -t 4 --fstats --vcf --structure --write-single-snp -O output_directory

- **Applies stricter filters:**
- **Minor allele frequency (MAF) ≥ 0.05 (removes rare alleles).**
- **Max observed heterozygosity ≤ 0.7 (eliminates highly heterozygous sites).**
- **Minimum 80% of individuals per population required.**
- **Outputs results to `output_high_quality/`.**

mkdir -p output_high_quality

populations -P gstacks_output -M population_map.txt -t 4 \

--fstats --vcf --structure --plink --write-single-snp \

--min-maf 0.05 --max-obs-het 0.7 --min-samples-per-pop 0.8 \

-O output_high_quality

- **Allows more variants:**
- **Minor allele frequency (MAF) ≥ 0.01 (retains rare variants).**
- **Max observed heterozygosity ≤ 0.9 (allows more heterozygous sites).**
- **Minimum 50% of individuals per population required.**
- **Saves results in output_relaxed/.**

mkdir -p output_relaxed

```
populations -P gstacks_output -M population_map.txt -t 4 \
  --fstats --vcf --structure --plink --write-single-snp \
  --min-maf 0.01 --max-obs-het 0.9 --min-samples-per-pop 0.5 \
 -O output_relaxed
```

Results:

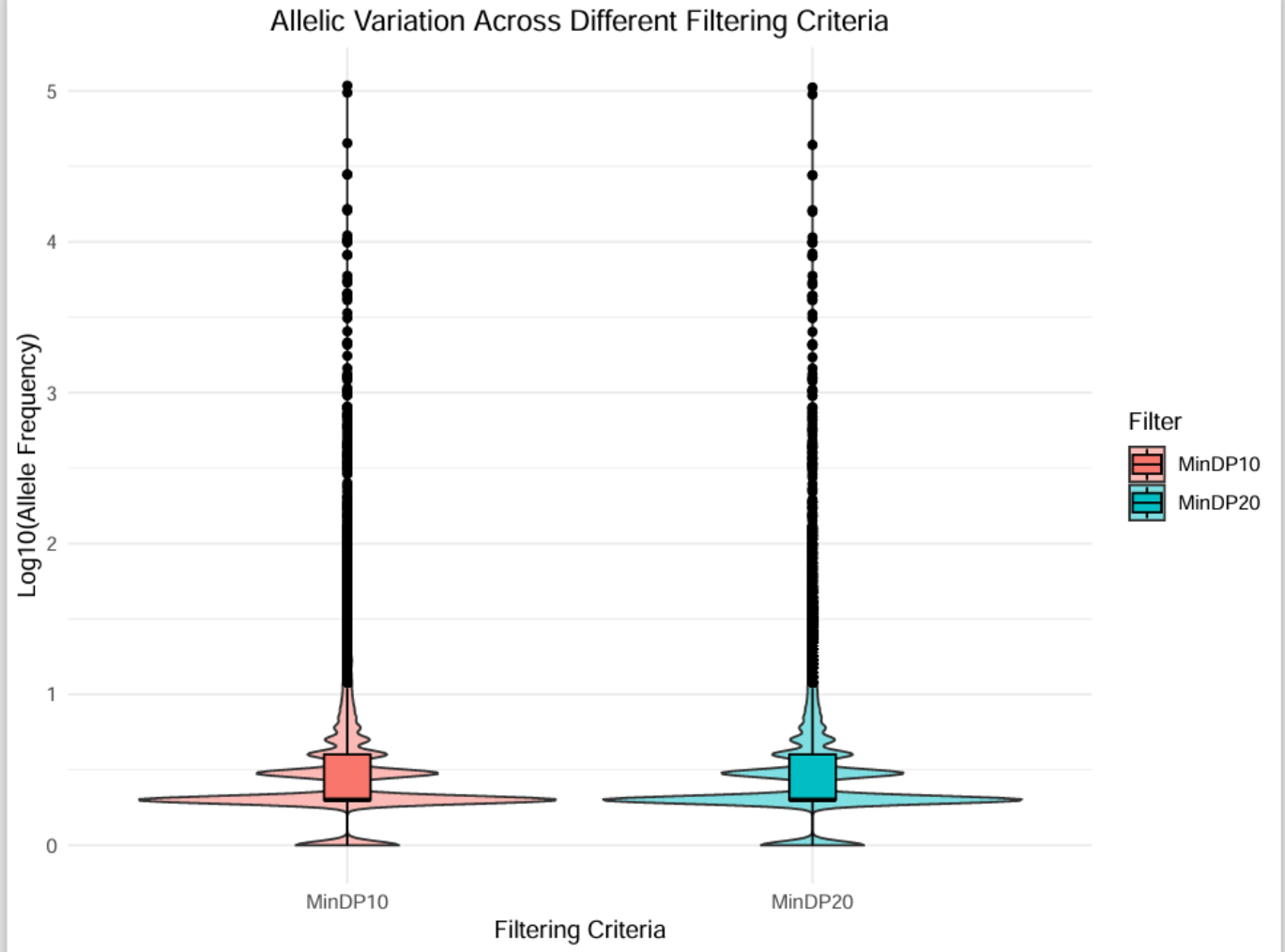

Figure 1:

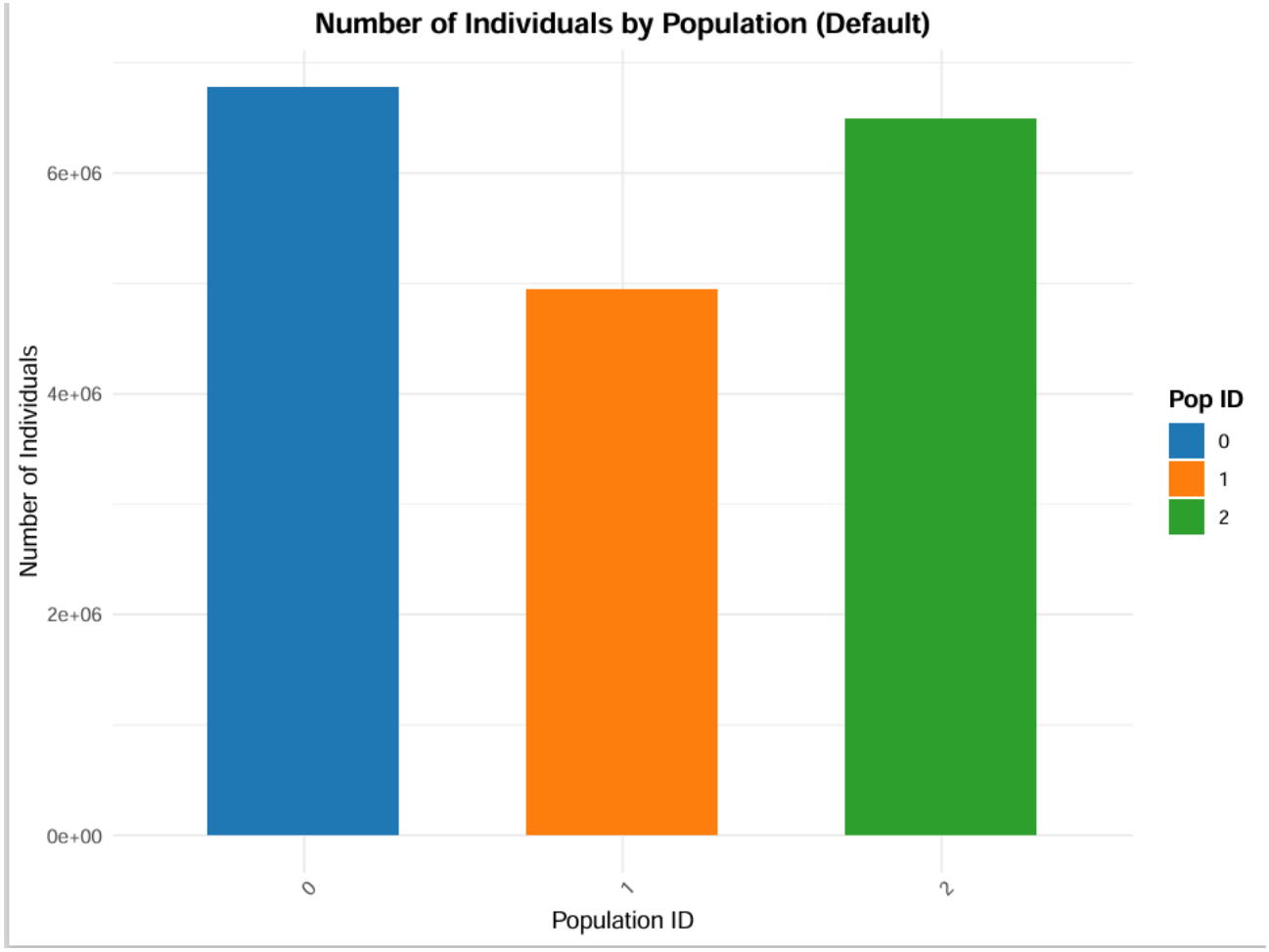
Number of Individuals by Population (Default)
Number of Individuals
6e+06
4e+06
2e+06
0e+00
0
1
2
Population ID
Pop ID
0
1
2

Figure 2:

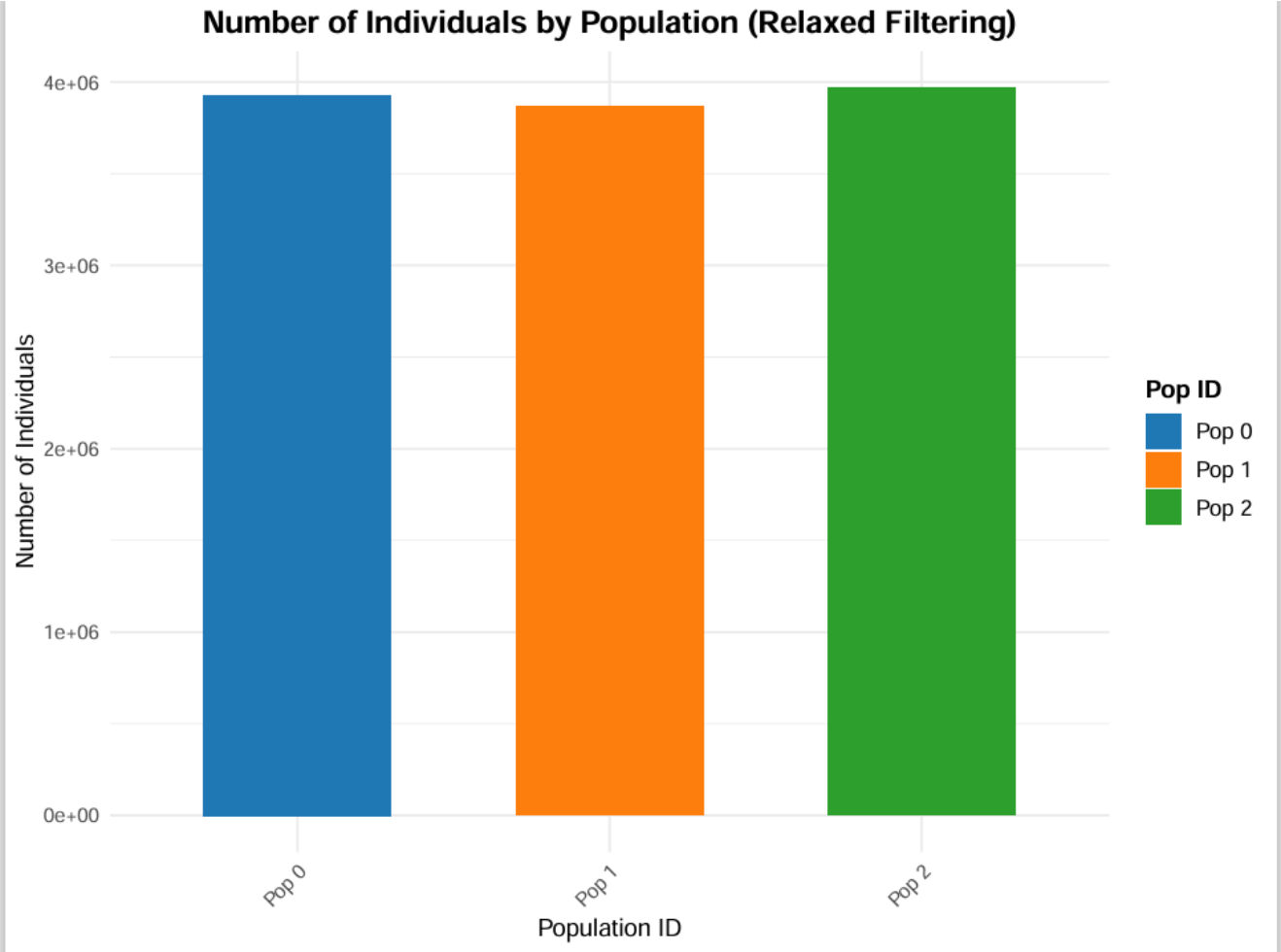
Number of Individuals by Population (Relaxed Filtering)
4e+06
3e+06
2e+06
1e+06
0e+00
Number of Individuals
Pop 0
Pop 1
Pop 2
Population ID
Pop ID
Pop 0
Pop 1
Pop 2

Figure 3:

Discussion:

The method applied captures allelic variation effectively however, perhaps raising the threshold (DP>20) reduced the number of low-frequency alleles. In figure 1, the violin plus boxplot represents the allelic variation across two different filtering criteria (DP>10 vs DP>20). The visual data depicted a large concentration of low-frequency alleles with high-frequency outliers. The boxplot demonstrated the central tendency and spread of the data. The two filtering criteria used depicted similar distribution, however, DP>20, has a slightly narrower distribution in the lower frequency alleles, this maybe indicative of having a stricter filtering criterion to remove false positive variant alleles. The use of different thresholds demonstrates, stringiest filtering criteria removes potentially false positives variants while maintaining distribution patterns, perhaps providing an accurate representation of allelic variations.

The population differentiation was compared between two thresholds, one was the default and the other a relaxed filtering criterion. Figures 2 and 3 both demonstrate similar number of individuals within each criterion. Under the default filtering, Population (Pop) 1 has fewer individuals than Pop 0 and 2, however, in the relaxed plot Pop1 catches up, indicating that more individuals were retained. In the default plot Pop 0 and 2 had more individuals in comparison to relaxed plot where the number of individuals were lower and consistent across all populations. Overall, the relative proportions remained consistent, demonstrating that filtering threshold did not drastically change population representation.

In conclusion, methods applied to measure allelic variation and population differentiation were effective, however, the choice of filtering parameters do impact the allelic diversity and

individual retention and understanding which parameters to use to obtain accurate results is essential.