**Command lines to make a consensus sequence and extract fragments containing 5 or more SNPs using a VCF file from reference genome GCA\_002763495.2\_F\_hepatica\_1.0.allpaths.pg\_genomic.fna**

1. First upload reference genome GCA\_002763495.2\_F\_hepatica\_1.0.allpaths.pg and merge read files generated by FASTp form Illumina reads of the whole genome of samples MR1 1/3\_no and VPC1 1/3 to HPC cluster University of Surrey.

# Change directory to the specified path

cd /parallel\_scratch/ma04426/Ref\ assembly\ F\_hepatica\_1.0.allpaths/

# Build Bowtie 2 index using the updated reference genome file

bowtie2-build GCA\_002763495.2\_F\_hepatica\_1.0.allpaths.pg\_genomic.fna genome\_index

1. **# Align the merged reads to the reference genome using Bowtie 2**

bowtie2 -x genome\_index \

-U MCR1\_1\_3\_no\_merged.fastq.gz \

-S output.sam \

--threads 4

Then rename the sam file produced (MR13\_no\_USA.sam)

1. **# Align the merged reads to the reference genome using Bowtie 2**

bowtie2 -x genome\_index \

-U VPC1\_1\_3\_merged.fastq.gz \

-S output.sam \

--threads 4

Then rename the sam file produced (VPC1\_13\_USA.sam)

1. **Command line for converting .sam file to .bam file**

# Change the directory to the specified path

cd /parallel\_scratch/ma04426/Ref\ assembly\ F\_hepatica\_1.0.allpaths/

samtools view -bS "MR13\_no\_USA.sam" > "MR13\_no\_USA.bam"

samtools view -bS "VPC1\_13\_USA.sam" > "VPC1\_13\_USA.bam"

1. **First, generate a pileup file from .bam files:**

**First, Sort the BAM file from both samples by using the following command lines**

# Change directory to the specified path

cd /parallel\_scratch/ma04426/Ref\ assembly\ F\_hepatica\_1.0.allpaths/

# Sort the BAM file

samtools sort "MR13\_no\_USA.bam" -o "MR13\_no\_USA.sorted.bam"

# Sort the BAM file

samtools sort "VPC1\_13\_USA.sam" -o "VPC1\_13\_USA.sorted.bam"

1. **# Index the first sorted BAM files**

# Index the sorted BAM file (MR13\_no\_USA.sorted.bam)

samtools index "MR13\_no\_USA.sorted.bam"

# Index the sorted BAM file (VPC1\_13\_USA.sorted.bam)

samtools index "VPC1\_13\_USA.sorted.bam"

1. **# Pileup for MR13\_no\_USA.sorted.bam**

samtools mpileup -f "GCA\_002763495.2\_F\_hepatica\_1.0.allpaths.pg\_genomic.fna" "MR13\_no\_USA.sorted.bam" > "MR13\_no\_USA.pileup"

**# Pileup for VPC1\_13\_USA.sorted.bam**

samtools mpileup -f "GCA\_002763495.2\_F\_hepatica\_1.0.allpaths.pg\_genomic.fna" "VPC1\_13\_USA.sorted.bam" > "VPC1\_13\_USA.pileup"

1. **Generating a consensus sequence from two samples, VPC1 1/3 and MR1 1/3\_no with the USA reference genome GCA\_002763495.2\_F\_hepatica\_1.0.allpaths.pg\_genomic.fna**

# Call variants for VPC1\_13\_USA.sorted.bam

bcftools mpileup -Ou -f GCA\_002763495.2\_F\_hepatica\_1.0.allpaths.pg\_genomic.fna VPC1\_13\_USA.sorted.bam | bcftools call -mv -Ov -o VPC1\_13\_USA.vcf

# Call variants for MR13\_no\_USA.sorted.bam

bcftools mpileup -Ou -f GCA\_002763495.2\_F\_hepatica\_1.0.allpaths.pg\_genomic.fna MR13\_no\_USA.sorted.bam | bcftools call -mv -Ov -o MR13\_no\_USA.vcf

**Explanation of the command:**

bcftools mpileup: This command generates a pileup from a sorted BAM file. A pileup is a compacted representation of the alignment information at each position in the reference genome.

-Ov: Specifies the output format (v stands for uncompressed BCF).

-f GCA\_002763495.2\_F\_hepatica\_1.0.allpaths.pg\_genomic.fna: Specifies the reference genome file in FASTA format.

|: The pipe (|) symbol is used to redirect the output of the first command as the input to the next command.

**bcftools call: This command is used to call variants from the pileup.**

-mv: Options for variant calling:

m: Multi-allelic calling.

u: Output variant sites only, excluding reference positions.

-Ou: Specifies the output format as a VCF file.

-o VPC1\_13\_USA.vcf: Specifies the output file name as VPC1\_13\_USA.vcf.

1. **Compress VCF files**

**Compress VCF Files using bgzip:**

# Compress MR13\_no\_USA.vcf using bgzip

bgzip MR13\_no\_USA.vcf

**# Compress VPC1\_13\_USA.vcf using bgzip**

bgzip VPC1\_13\_USA.vcf

1. **First, index the VCF File with GATK 4.4.0.0**

java -Dsamjdk.use\_async\_io\_read\_samtools=false \

-Dsamjdk.use\_async\_io\_write\_samtools=true \

-Dsamjdk.use\_async\_io\_write\_tribble=false \

-Dsamjdk.compression\_level=2 \

-jar /opt/software/pkgs/GATK/4.4.0.0-GCCcore-12.2.0-Java-17/gatk-package-4.4.0.0-local.jar \

IndexFeatureFile -I VPC1\_13\_USA.vcf.gz

java -Dsamjdk.use\_async\_io\_read\_samtools=false \

-Dsamjdk.use\_async\_io\_write\_samtools=true \

-Dsamjdk.use\_async\_io\_write\_tribble=false \

-Dsamjdk.compression\_level=2 \

-jar /opt/software/pkgs/GATK/4.4.0.0-GCCcore-12.2.0-Java-17/gatk-package-4.4.0.0-local.jar \

IndexFeatureFile -I MR13\_no\_USA.vcf.gz

1. **Then filter variants with GATK 4.4.0.0**

# For MR13\_no\_USA.vcf.gz

gatk VariantFiltration -V MR13\_no\_USA.vcf.gz -O MR13\_3filtered\_variants.vcf.gz --filter-expression "QD < 2.0 || FS > 60.0 || MQ < 40.0" --filter-name "basic\_filter"

# For VPC1\_13\_USA.vcf.gz

gatk VariantFiltration -V VPC1\_13\_USA.vcf.gz -O VPC1\_3filtered\_variants.vcf.gz --filter-expression "QD < 2.0 || FS > 60.0 || MQ < 40.0" --filter-name "basic\_filter"

1. **Then, Create a PASS-Only VCF files for both samples with GATK 4.4.0.0**

# For MR13\_no\_USA.vcf.gz

java -Dsamjdk.use\_async\_io\_read\_samtools=false \

-Dsamjdk.use\_async\_io\_write\_samtools=true \

-Dsamjdk.use\_async\_io\_write\_tribble=false \

-Dsamjdk.compression\_level=2 \

-jar /opt/software/pkgs/GATK/4.4.0.0-GCCcore-12.2.0-Java-17/gatk-package-4.4.0.0-local.jar \

SelectVariants -V /parallel\_scratch/ma04426/Ref\ assembly\ F\_hepatica\_1.0.allpaths/MR13\_3filtered\_variants.vcf.gz --exclude-filtered true -O /parallel\_scratch/ma04426/Ref\ assembly\ F\_hepatica\_1.0.allpaths/MR13\_3pass\_variants.vcf.gz

# For VPC1\_13\_USA.vcf.gz

java -Dsamjdk.use\_async\_io\_read\_samtools=false \

-Dsamjdk.use\_async\_io\_write\_samtools=true \

-Dsamjdk.use\_async\_io\_write\_tribble=false \

-Dsamjdk.compression\_level=2 \

-jar /opt/software/pkgs/GATK/4.4.0.0-GCCcore-12.2.0-Java-17/gatk-package-4.4.0.0-local.jar \

SelectVariants -V /parallel\_scratch/ma04426/Ref\ assembly\ F\_hepatica\_1.0.allpaths/VPC1\_3filtered\_variants.vcf.gz --exclude-filtered true -O /parallel\_scratch/ma04426/Ref\ assembly\ F\_hepatica\_1.0.allpaths/VPC1\_3pass\_variants.vcf.gz

1. **Merge VCF Files using module load GCC/12.2.0 BCFtools/1.17**

bcftools merge MR13\_3pass\_variants.vcf.gz VPC1\_3pass\_variants.vcf.gz -o merged\_variants.vcf

1. **Compress the VCF File with bgzip:**

bgzip merged\_variants.vcf

1. **Sort and Index the Merged Variants:**

# Sort the gzipped VCF file and save the sorted VCF file

bcftools sort merged\_variants.vcf.gz -o sorted\_merged\_variants.vcf

bgzip sorted\_merged\_variants.vcf

# Index the sorted VCF file

bcftools index sorted\_merged\_variants.vcf.gz

1. **Create the Consensus Sequence**

Index the Compressed VCF File

tabix -p vcf merged\_variants.vcf.gz

**Generate the Consensus Sequence:**

bcftools consensus -f GCA\_002763495.2\_F\_hepatica\_1.0.allpaths.pg\_genomic.fna -o consensus\_sequence.fa merged\_variants.vcf.gz

1. **You can see and examine the consensus sequence in the IGV viewer**
2. **Extract fragments from the reference genome** GCA\_002763495.2\_F\_hepatica\_1.0.allpaths.pg\_genomic.fna using the merged VCF file with threshold limits 5 or more consecutive or non-consecutive SNPs

To do this, first produce a BED file from the merged VCF file with threshold limits 5 or more consecutive or non-consecutive SNPs

bcftools query -f '%CHROM\t%POS\n' merged\_variants.vcf.gz | \

awk -v threshold=5 '{

if (last\_chrom != $1) {

last\_chrom = $1

last\_pos = $2

snp\_count = 1

} else {

if ($2 <= last\_pos + 1) {

snp\_count++

} else {

snp\_count = 1

}

}

last\_pos = $2

if (snp\_count >= threshold) {

print $1, $2 - snp\_count + 1, $2

}

}' > consecutive\_5\_or\_more\_snps.bed

1. **Manipulation of BED file and extracting fragments**

Modify the bed file by expanding numbers in columns two and three by 100 on each side by subtracting 100 in column 2 and adding 100 from column 3 with the following command

awk -v left=100 -v right=100 'BEGIN{OFS="\t"} { $2 -= left; $3 += right; print }' consecutive\_5\_or\_more\_snps.bed > modified\_consecutive\_snps.bed

1. **Extract fragments from the reference sequence with BEDtools**

bedtools getfasta -fi GCA\_002763495.2\_F\_hepatica\_1.0.allpaths.pg\_genomic.fna -bed modified\_consecutive\_snps.bed -name -fo extracted\_sequence\_fragments.fasta

Visualise the extracted fasta file containing 5 or more SNPs in Geneious