**Botany 2022 Genome Assembly**

This module was written by Jacob Landis ([jbl256@cornell.edu](mailto:jbl256@cornell.edu)) for the *de novo* genome assembly and annotation workshop for the Virtual Botany 2020 meeting. All of the data used here are publicly available from the Short Read Archive and other data repositories. This module uses the CyVerse framework and a virtual machine that has been created with all of the necessary programs already installed. The associated presentations and analyses scripts for the entire module can be found on GitHub (<https://github.com/bcbc-group/Botany2020NMGWorkshop>). If you have questions about anything, especially trying to run some of these with your own data and on your local server/cluster, please contact me.

Now that you have a genome assembly, even a draft version, there are many things that you can do with this. One is to use it in population genetic studies as a reference to call SNPs using a RAD-Seq, Hyb-Seq, or whole genome resequencing approaches. Because we are not generating any new data here, we are going to be using some RNA-Seq data to call SNPs using the Stacks pipeline. For nonRAD-Seq data other programs such as GATK, FreeBayes, or Samtools might be better to use but they can be more difficult to learn in a short-time frame and can take a lot longer to run. However, with a reference approach, all these methods share an initial step, mapping the cleaned reads of interest to the reference genome.

**SNP Calling**

Before read mapping, the first thing to do is download the data of interest and/or make sure the fastq files you have are cleaned and ready to be used. To download raw sequencing data, one of the easiest methods is to use the ENA database and wget and then to rename the file to something that is easier to read (the data has already been downloaded for the workshop, but here is an example of what you would do on your own) –

/data-store/iplant/home/jbl256/Botany2022

mkdir SNP\_calling

cd SNP\_calling

wget <ftp://ftp>.sra.ebi.ac.uk/vol1/fastq/SRR106/046/SRR10676746/SRR10676746.fastq.gz

mv SRR10676746.fastq.gz Ugibba\_bladderR1.fastq.gz

To clean up the data, either your own or downloaded data sets, the easiest and fastest way is using the program fastp. The parameters here are for single end reads input, output, Illumina adapater sequence, minimum quality score, minimum length to keep, and number of threads. This step has already been done on the data we will use, so no need to do it again here.

/home/user/fastp -i Ugibba\_bladderR1.fastq.gz -o Ugibba\_bladder\_cleaned\_R1.fastq.gz -z 4 –adapter\_sequence=AGATCGGAAGAGCACACGTCTGAACTCCAGTCA –adapter\_sequence\_r2=AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT -q 20 –length\_required 100 –thread 6

Now that you have cleaned reads ready to go, we will be using the program BWA to map the reads to the reference, specifically using the bwa mem function. To save some time, we will use a pruned genome file that only contains four chromosomes instead of the full assembly. The first step is to index the reference fasta file to make it easier to navigate and locate read locations:

scp /data-store/iplant/home/jbl256/Botany2022/Completed\_assemblies/Ugibba\_pruned\_assembly.fasta /data-store/iplant/home/jbl256/Botany2022/SNP\_calling

export PATH=/home/user/bwa:$PATH

bwa index Ugibba\_pruned\_assembly.fasta

With the reference now indexed, one at a time we will map the reads from the RNA-Seq data sets to the reference. When using multiple samples, it is advisable to use read group information to make it easier to filter or subset the data later on. The read group information consists of four different types of information:

Read group information

ID: is unique identifier of the samples, for now doing the sample name and the barcode info

SM: is the sample name

PL: is the sequencing equipment, in almost all cases this will be Illumina

PU: is the run identifier, the lane, followed by the specific barcode of the sample

LB: is the library count or other relevant information

The mapping script below calls the program, enforces read group information, specifies the reference fasta file, the sequencing data, and then the output.

bwa mem -t 6 -R "@RG\tID:bladderR1\tSM:Rep1\tPL:HiSeq\tPU:HTNMKDSXX\tLB:RNA-Seq" Ugibba\_pruned\_assembly.fasta Ugibba\_bladder\_cleaned\_R1.fastq.gz > Ugibba\_bladderR1.sam

The output file is a SAM file. To save memory requirements, we will convert this to a binary format and then sort with Samtools. You can do this individually for each file, but to make it easier we have a script set up to run through each of RNA-Seq data files, map to the reference, convert to binary, and then sort. At the end all of your sorted BAM files can be found in a new folder “sorted\_bam”. To run this script, just type:

/home/user/work/data/iplant/home/shared/Botany2020NMGWorkshop/scripts/BWA\_Ugibba.sh

With the BAM files for all the samples, we are ready to start calling SNPs. One additional piece of information that we need to supply is a population map for Stacks. This lets the program know what samples to use/expect, and if any are found in the same population. This doesn’t impact the actual SNP calling but will impact the exporting of some result files. The population map looks like this:

Ugibba\_bladderR1 bladder

Ugibba\_rhizoidR1 rhizoid

We will use the ref\_map.pl wrapper to invoke all the necessary components of Stacks. See their online manual for a better idea of what each step is doing. First we make a new directory for all the output files, then specify the folder where the BAM files to use are, the population map, and then the output directory:

cd /data-store/iplant/home/jbl256/Botany2022

mkdir ref\_SNPs

/home/user/stacks-2.62/scripts/ref\_map.pl --samples /home/user/work/data/iplant/home/shared/Botany2020NMGWorkshop/Genome\_assembly/sorted\_bam --popmap /home/user/work/data/iplant/home/shared/Botany2020NMGWorkshop/Genome\_assembly/SNP\_calling/population\_map.txt -o /data-store/iplant/home/jbl256/Botany2022/ref\_SNPs/ -T 6

Once the run finish, we will need to export the results to a VCF file using the Stacks function populations:

/home/user/stacks-2.62/populations --batch\_size 1 -P ref\_SNPs/ -M /home/user/work/data/iplant/home/shared/Botany2020NMGWorkshop/Genome\_assembly/SNP\_calling/population\_map.txt -t 6 --ordered\_export --vcf

The exported file is unfiltered and ordered based on chromosome and position. Stacks does implement some filtering capacity, but those methods are a bit limited. To better filter the SNP data set, we will use VCFtools. We will need to install the program first since it wasn’t installed previously. First navigate to your scratch directory and follow the steps below:

/home/user/vcftools-vcftools-581c231/bin/vcftools --help

to filter at loci with more than 40% missing data, keep only sites with a coverage between 3-100, keep a minor allele frequency of at least 0.05, and then recode the header information after filtering.

cd /data-store/iplant/home/jbl256/Botany2022/ref\_SNPs

/home/user/vcftools-vcftools-581c231/bin/vcftools --vcf /home/user/work/data/iplant/home/shared/Botany2020NMGWorkshop/Genome\_assembly/SNP\_calling/populations.snps.vcf --max-missing 0.6 --min-meanDP 3 --max-meanDP 100 --maf 0.05 --mac 3 --recode --recode-INFO-all --out Ugibba\_SNPs\_filtered

#initial pass throwing out all SNPs that are missing 60%

/home/user/vcftools-vcftools-581c231/bin/vcftools --vcf /home/user/work/data/iplant/home/shared/Botany2020NMGWorkshop/Genome\_assembly/SNP\_calling/populations.snps.vcf --max-missing 0.4 --min-alleles 2 --max-alleles 2 --recode --recode-INFO-all --out Ugibba\_first\_pass

#gives missing proportion of loci for each individual

/home/user/vcftools-vcftools-581c231/bin/vcftools --vcf Ugibba\_first\_pass.recode.vcf --missing-indv

#average depth for each individual

/home/user/vcftools-vcftools-581c231/bin/vcftools --vcf Ugibba\_first\_pass.recode.vcf --depth

#observed and expected heterozygosity

/home/user/vcftools-vcftools-581c231/bin/vcftools --vcf Ugibba\_first\_pass.recode.vcf --het

#create a list of individuals with at least 50% missing data

awk '$5 > 0.75' out.imiss | cut -f1 > lowDP50.indv

/home/user/vcftools-vcftools-581c231/bin/vcftools --vcf Ugibba\_first\_pass.recode.vcf --max-missing 0.4 --remove lowDP50.indv --recode --recode-INFO-all --out Ugibba\_filtered\_SNPs

#gives missing proportion of loci for each individual

/home/user/vcftools-vcftools-581c231/bin/vcftools --vcf Ugibba\_filtered\_SNPs.recode.vcf --missing-indv

#average depth for each individual

/home/user/vcftools-vcftools-581c231/bin/vcftools --vcf Ugibba\_filtered\_SNPs.recode.vcf --depth

#observed and expected heterozygosity

/home/user/vcftools-vcftools-581c231/bin/vcftools --vcf Ugibba\_filtered\_SNPs.recode.vcf –het