SNP Analyses in the Discovery Environment

Put together by Jacob Landis for the CyVerse Webinar "Got Variants? Downstream Analyses for PopGen and Evolution Studies" on February 5th, 2021

This is a modified version of the Discovery Environment tutorial that is intended to be done on your own local machine or local server predominately by the command line. Some programs such as BEAUTi, SplitsTree, and Tassel have a GUI (Graphical User Interface) component. All of the main steps are still included, but you will need to make sure that the correct programs are installed. Additionally, I have given the commands for some analyses in this document, but they can also be found in shell scripts in the appropriate folders. Make sure to change to the appropriate directory before trying to implement any of the commands. Some of the files have been gzipped before uploading to GitHub to save space. If you find a gzip file (.gz) please make sure to gunzip before proceeding.

File conversion

Stacks populations (stacks 2.53 populations)

#after vcftools filters individuals

populations --batch_size 1 -V Washingtonia_test_samples.recode.vcf -O data_set_individuals -M pop_map_individuals.txt -t 2 --ordered-export --vcf --genepop --structure

VCF2Phylip

This program converts our VCF to Nexus and Fasta easily, which are necessary formats for some of the downstream analyes.

python3 vcf2phylip.py -i Washingtonia_test_samples.recode.p.snps.vcf -n -f

SNPhylo

This program will generate a Maximum-likelihood tree from our data set. This file was previously modified to make this run easier by making sure that the first column of the VCF file has only numbers to designate chromosomes. In your data, this may need to be modified. Any letters/words in the contig/chromosome names will cause problems. The number of chromosomes/contigs doesn't impact the run, however the default is set to 22 (autosomes in humans). The maximum number can be modified with the -a flag (or The_number_of_the_last_autosome in the DE App)

#ML tree from VCF data

snphylo.sh -v Samples.vcf -p 95 -c 5 -l 1.0 -m 0.05 -M 0.8 -P ML_tree -a 22

Once the run completes, download/open the file “snphylo.output.ml.tree” in FigTree or similar viewing program to export as a PDF.

SNAPP with Beast2

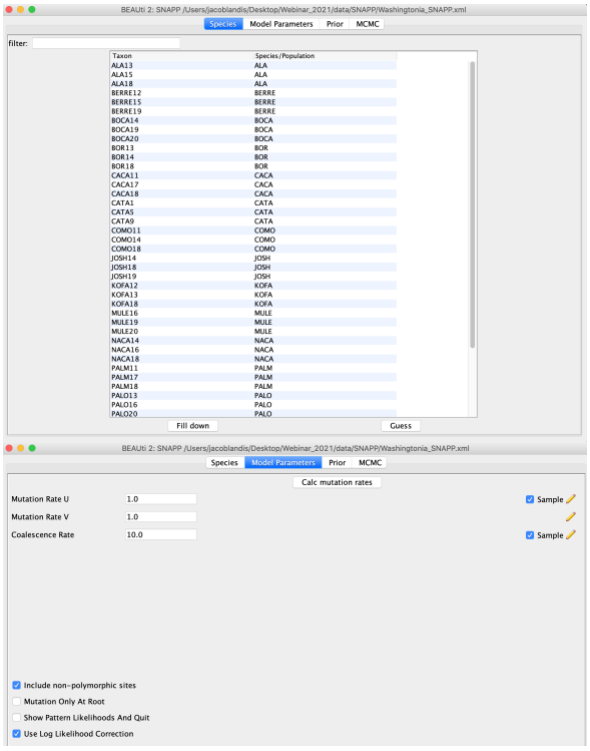
This approach takes a Bayesian framework to estimate a species tree based on the SNP data. The actual tree inference is quite computationally intensive and can take a long time to run with lots of samples and many SNPs. The analysis ignores SNPs with missing data, so I would suggest pruning the input file even more to only include SNPs with no missing data and thinned to reduce the total number of SNPs. This has been done previously in VCFtools with following command:

vcftools --vcf Samples.vcf --keep SNAPP.txt --max-missing 1.0 --thin 5000 --recode --recode-INFO-all --out Samples_SNAPP

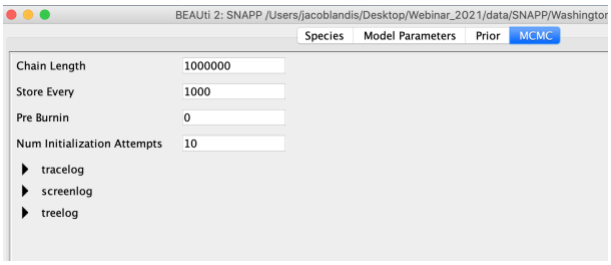
If you do this yourself, make sure to prune the VCF, then run this pruned file through vcf2phylip to generate a new fasta file.

Also important, make sure to use the newest version of Beast 2.6.3. Older versions look for different versions of the dependencies, and things will fail. https://www.beast2.org/

Open BEAUTi on your local machine.
Click Manage Packages -> Install SNAPP if it is not already.
Template -> SNAPP
File -> Add Alignment -> Washingtonia_test_SNAPP.recode.min4.fasta
Need to specify species/populations, to do this delete the numbers after each accession. The populations are specified as part of the taxon name.
Under the Model Parameters tab click “Calc mutation rates”

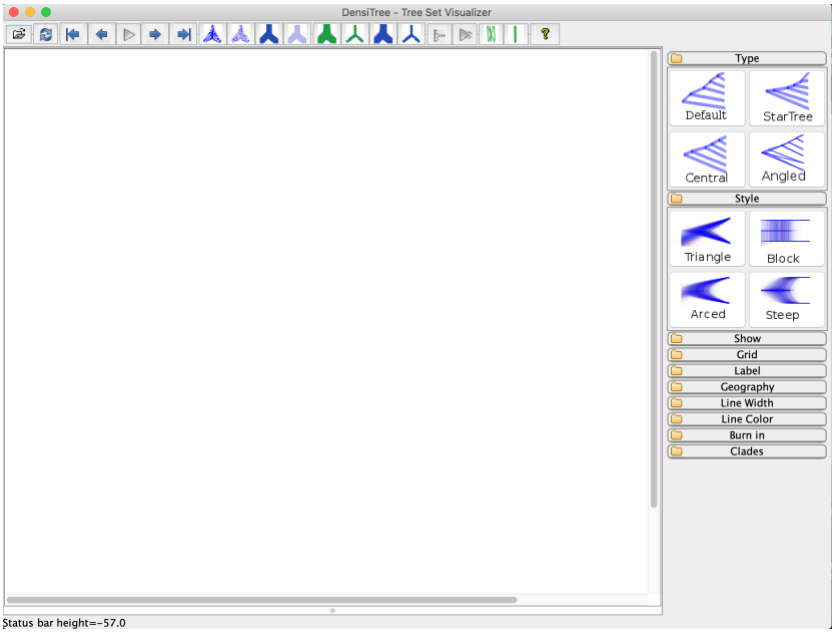


MCMC tab, set to 100,000 generations as a test (full runs will take much longer)
Save XML file.

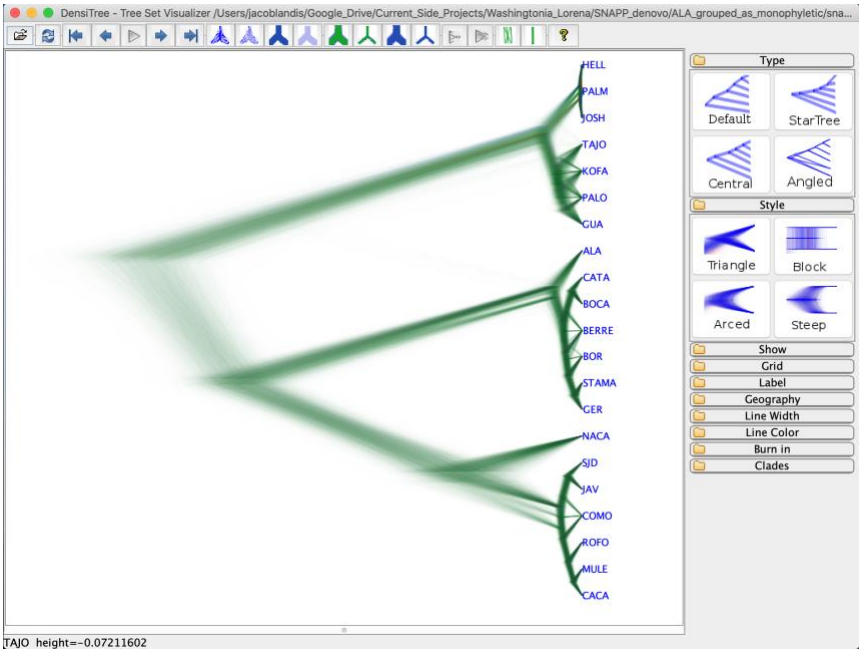


Then launch the created XML file in Beast
java -jar beast.jar -threads 24 Washingtonia_SNAPP_2.6.3.xml

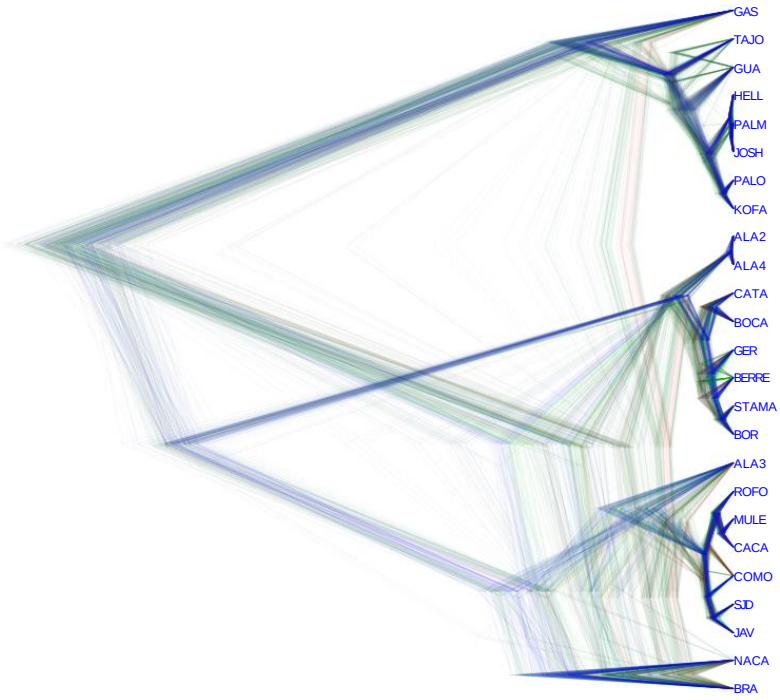
After the run completes, you can open the resulting snapp.trees file with DensiTree on your local machine.



File -> Load -> snapp.trees



If the tree looks messy (see below) compared to the one above, the analysis needs to run for longer. The snapp.log can be opened in Tracer to explore convergence. As with most Beast analyses, an ESS value above 200 is considered sufficient.



SplitsTree

This approach is often considered better for interbreeding populations since it does not rely on the assumption of a bifurcating relationship. This analysis will also be conducted on your local machine, though it does complete very quickly even with larger data sets. The input for this is a

nexus file created from vcf2phylip. This analysis does not require the further pruned data set, but either will work.

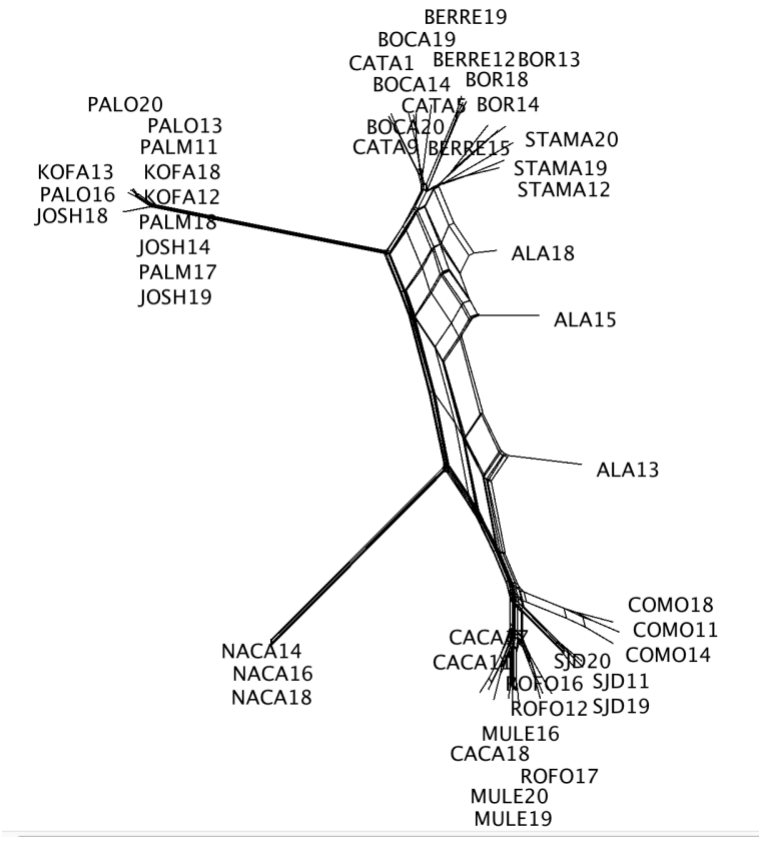
Download the appropriate version for your local machine

- SplitsTree4: https://software-ab.informatik.uni-tuebingen.de/download/splitstree4/welcome.html
- SplitsTree5: https://software-ab.informatik.uni-tuebingen.de/download/splitstree5/welcome.html

Open the Graphical User Interface and load the nexus file created earlier.

File -> Open -> Washingtonia_test_samples.recode.p.snps.min4.nexus

Many options to explore, but the basic Neighbor Net is now shown.



PCA (Rstudio)

This analysis will take the full SNP data set and work to find the major differences between the samples and highlight the amount of variation explained in the first couple principal components.

You should have the following files in the current working directory folder:

- Data file in VCF format: Washingtonia_test_samples.recode.p.snps.vcf
- Main Rscript with all commands: **SNPRelate.R**
- A population map assigning each sample to a population: popmap.txt
- A file with just the populations in the same order as the population map file for color coding the plots: pop.txt

Run through the R script. The script is set up to save a PDF of each .

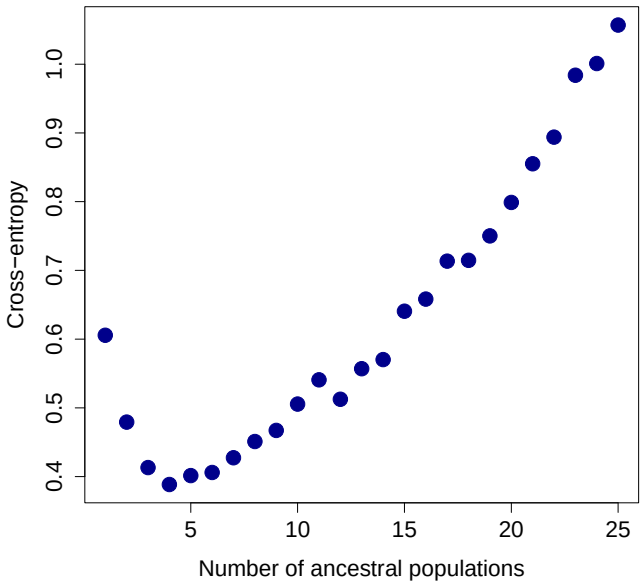
Structure (Rstudio)

This analysis will determine the most likely number of ancestral populations based on your current data and show which individuals/populations may be admixed. There are many ways to do this, but we will be using the LEA package in R. One key piece of information, we are using the Structure file generated in the Stacks earlier. However, that file has two header lines that have issues being read by LEA. For ease, we first deleted those two extra rows (no data was deleted). If you try to run this on your own data, you will likely need to do the same.

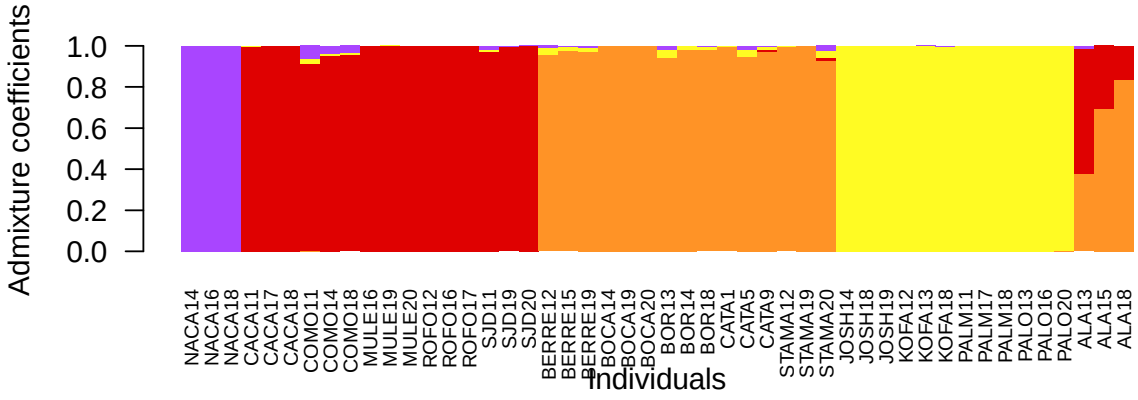
There are four files that are needed in this folder:

- Data file created from the Stacks populations program: Washingtonia_test_samples.recode.p.structure
- The main **Structure.R** script with all the commands
- Two additional R files: POPSutilities.R and Conversion.R, which were downloaded from the LEA package

Run through the R script as you would on your local machine. The script is set up to save a PDF of each plot. In the end you will have a plot that will help determine the Best K for this data set with a cross-entropy plot. The low point on the graph is the best K value.



You will also be generating multiple Structure plots for different values of K. Here is an example from the Best K=4. Each time you run this analysis in R the plot may be slightly different, in some case the proportion of different admixed groups may change slightly, but the general pattern should be the same.



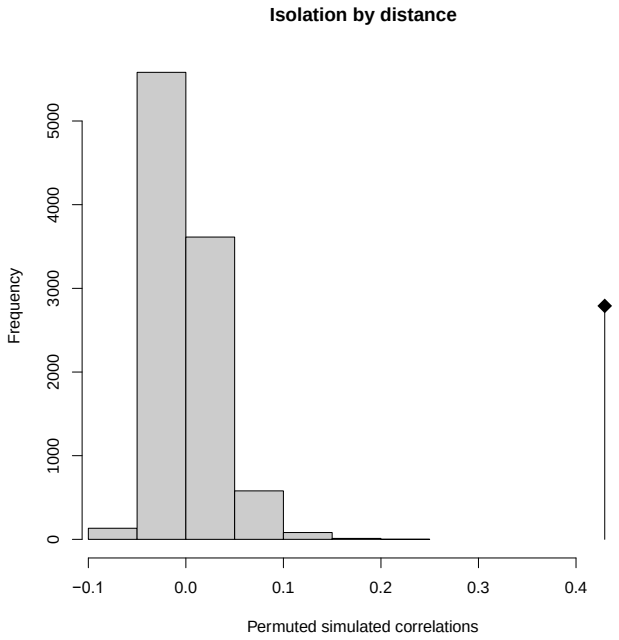
Isolation by Distance (IBD; Rstudio)

This can be calculated in Rstudio with the package adegenet being the main driver. The main pieces need for this include the genetic information and geographic coordinates. You should start another Rstudio instance with the IBD folder. One thing of note, here we are using the GenePop file created by Stacks earlier, however adegenet does not recognized .genepop, so the file has modified to end with .gen. The data in the file was not altered in anyway.

There are three files in this folder that you will need to run the analysis:

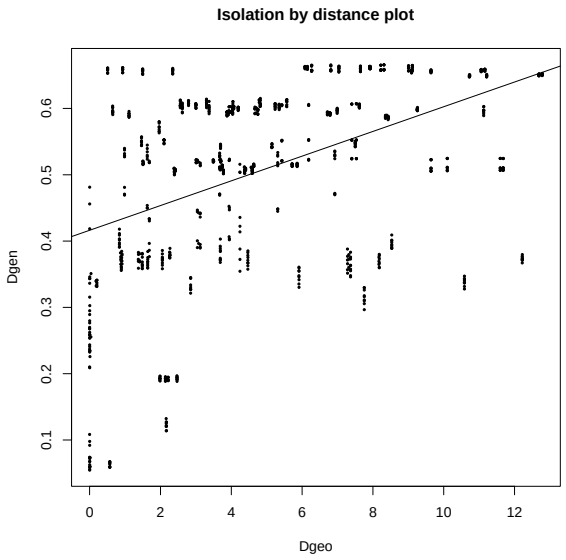
- Data file produced by Stacks populations (with a modified extension of .gen instead of .genpop): Washingtonia_test_samples.recode.p.snps.gen
- The Rscript with all the steps: **IBD.R**
- Latitude and Longitude values for each sample in decimal format: latlong.txt

Run through the R script as you would on your local machine. The script will test for IBD and do 10,000 permutations to test for significance. The script is set up to save a PDF of each plot. The following plots will be generated. The first is the observed value of IBD as well as the results from the permutation test, the observed value is the point of the diamond.



The specific p-value is not shown here, but can be found in the onscreen output from Rstudio. In this case, it was:
Monte-Carlo test
Observation: 0.4294366
Based on 10000 replicates
Simulated p-value: 9.999e-05

Plots of geographic distance vs genetic distance will also be plotted, one with colors showing differences and one without color. The best fit line is included.



Genome Wide Association

Tassel (Using your local machine)

There are many options for performing Genome Wide Association. All are slightly different in terms of the input data needed and how picky the programs are in formatting. One of the easiest to use is Tassel, though this does not have the same flexibility in plotting as other options do (it does however make plots for you). The input files are our VCF file for genetic information and a Phenotypes file with four traits. The Phenotypes file has a specific format needed for Tassel.

Download the newest version of Tassel: https://www.maizegenetics.net/tassel. For this tutorial we will be using the Graphical User Interface, however there was recently a released R wrapper for Tassel called rTassel (https://maize-genetics.github.io/rTASSEL/). Worth exploring as a potential option for those interested.

There are many tutorials online, but these steps below will work for the given example data.
Import data

Open as -> Best guess, sort positions, keep depth -> Select VCF with genetic data

Import trait data

Phenotype

GLM (General Linear Model)

-Select Sequence data, Analysis -> Relatedness -> Distance Matrix

-Select Distant Matrix, then Analysis -> Relatedness -> MDS

-Select MDS_PCs_Matrix, Sequence data, Phenotype data, then Data -> Intersect Join

Association test

-Select the new file you just created, Analysis -> Association -> GLM

Plotting

-Select GLM_Stats -> Results -> Manhattan Plot

MLM (Mixed Linear Model)

Produce kinship matrix

-Select Sequence file, then Analysis -> Relatedness -> Kinship -> default parameters

Association test

-Select sequence data and phenotype data, then Data -> Intersect Join

-Select joined data and Kinship matrix (Centered_IBS), then Analysis -> Association -> MLM -> Run analysis

Plotting

-Select MLM_stats -> Results -> Manhattan Plot

EMMAX with Plink

This method is often found in more recent publications. It calculates the Kinship matrix a bit differently from Tassel and only implements a Mixed Linear Model. It does not produce a Manhattan plot for you, but the files it produces can be loaded into R for plotting. This gives a bit more flexibility in terms of color scheme and other plotting options. Also, we will have to run each trait separately, whereas Tassel did all four traits with the same clicks.

#convert from VCF to Plink format

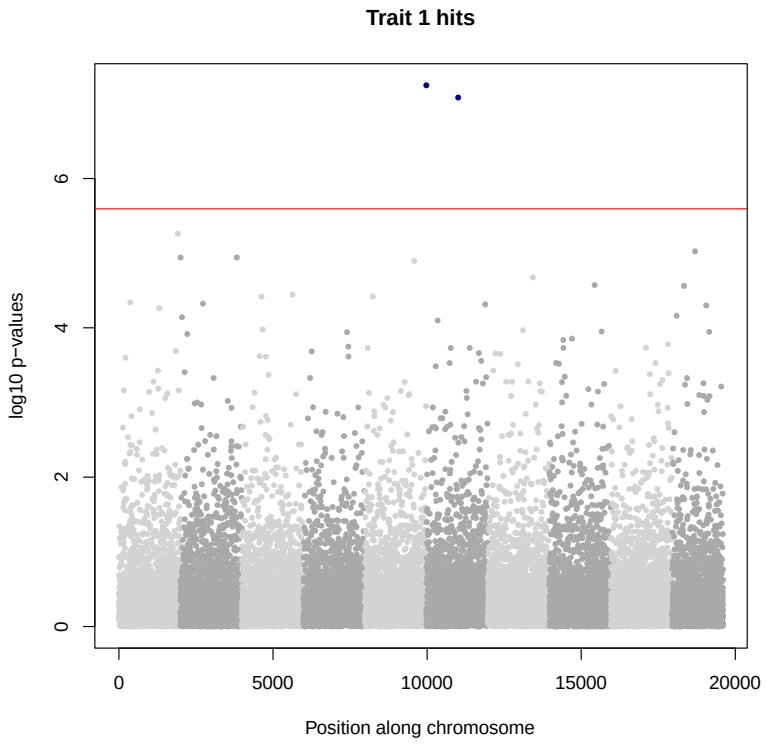
plink --vcf Washingtonia_modified_GWAS_genotype_data.vcf --maf 0.05 --double-id --recode 12 --out plink_filter --output-missing-genotype 0 --transpose --allow-extra-chr

After the run completes you will see the following files: .tped, .tfam, .nosex, .map, and .log. Some of these will be used for EMMAX directly.

Once that is done now we can use the EMMAX app for the analysis

The remaining steps for the EMMAX are in the shell script "**EMMAX.sh**" in the GWAS/EMMAX folder. This will walk through the remaining steps. The output files needed will be in the created GWAS_output folder. The Trait.ps files are what are needed for the plotting R **gwas_plot.R**. This script is modified to match the format of the .ps files and create Manhattan plots.

For Trait 1, the Manhattan plot looks like this with two SNPs above the significance threshold:


Trait 1 hits
log10 p−values
0
2
4
6
0
5000
10000
15000
20000
Position along chromosome