# 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

After signing into CyVerse (cyverse.org) click Launch the Discovery Environment. You will need to sign in again probably. On the left you should see three buttons: "Data", "Apps", and "Analyses". For this tutorial we will mostly be using the top two, but you can click the "Analyses" button at any time to see what is currently running on your account.

Click the "Data" button. Your username should be listed on top. Click on that, then the "analyses" folder. Click File ->New Folder to create a folder called Tutorial. This will be our main repository for this walkthrough. All of the files that you will need for this have previously been uploaded to a shared Example Data folder found in the following directory: "/iplant/home/shared/iplantcollaborative/example_data/SNPanalysis_webinar_spring21/Webinar_Feb2021". All of the data can also be downloaded directly from GitHub to use on your own machines (https://github.com/jblandis/CyVerse_Variant_Analyses). Each type of analysis has its own folder with the specific files you will need to complete that step. For every analysis you run, you will create a new folder for that run automatically and files will be put in there. Files can be moved around either in the DE or via Cyberduck. We are not going to be covering those specifics here since there are already resources on how to do that (https://learning.cyverse.org/projects/data_store_guide/en/latest/step1.html#:~:text=In%20the%20Cyberduck%20configuration%20window,bookmark%20in%20the%20Cyberduck%20window).

Almost every step in this tutorial has its own App, though some steps such as BEAUTi, SplitsTree, and Tassel, you will do on your local machines since those programs come with standalone GUIs (Graphical User Interface) and run very quickly on almost any platform. For these make sure you download the necessary data files from GitHub (see above). For the Apps in DE, I have included a picture of the App itself and the specific name to be found by searching in the Apps directory.

## File conversion

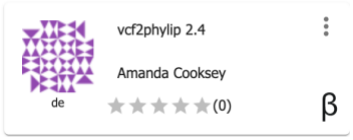
## Stacks populations (stacks 2.53 populations)


*Analysis* – Leave the name and output folders as default, or you can change them to make easier to find later

*Input* - Select VCF file from File Conversion folder "Washingtonia_test_samples.recode.vcf", and population map for individuals "pop_map_individuals.txt"

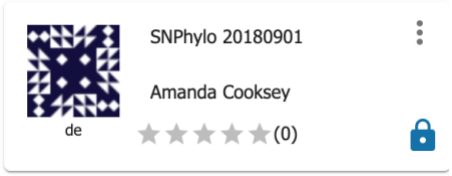
*Data Filtering* - We have already filtered this data set previously so we'll leave this option blank. If this is your first time creating a VCF file for a project you may want to consider adding in some options, though I prefer to do most of the filtering in VCFtools or BCFtools since both programs have a bit more functionality.
*File output options* - For this we want to select ordered export (sorted by chromosome and base pair position), and create a Structure, GenePop, and a new VCF file. Others can be done as necessary
Then hit *Launch analysis*

## VCF2Phylip (vcf2phylip 2.4)




This App converts our VCF to Nexus and Fasta easily, which are necessary formats for some of the downstream analyes.
*Input* - VCF file from file conversion folder "Washingtonia_test_samples.recode.vcf"
*Options* - select write Fasta, write Nexus
*Launch analysis*

## SNPhylo (SNPhylo 20180901)




This App 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)

*Input* – VCF file in the SNPhylo folder: Washingtonia_SNPhylo_dataset.recode.vcf
*Options* – listed as seen in the DE App, but the flags for command line are also included

Maximum_PLCS (-p): 95
Minimum_depth_of_coverage (-c) 2:
LD_threshold (-l): 1.0
MAF_threshold (-m): 0.05
Missing_rate (-M): 0.8
The_number_of_the_last_autosome (-a): 22

*Launch Analysis*

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

## SNAPP (local machine for generating XML but then Beast 2 App for running the analyses)

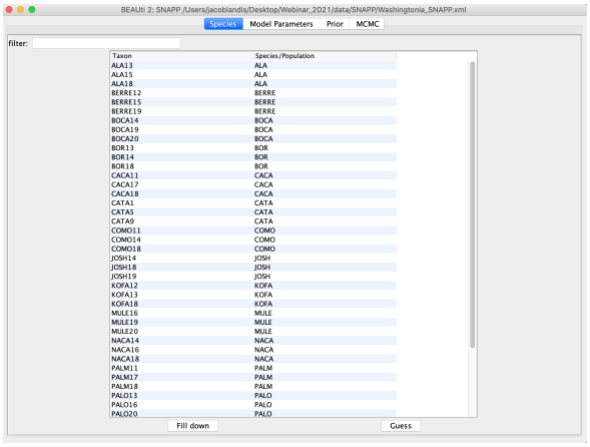
This approach takes a Bayesian framework to estimate a species tree based on the SNP data. The actual tree inference is quite computationally intensive and can take a long time to run with lots of samples and many SNPs. The analysis ignores SNPs with missing data, so I would suggest pruning the input 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:

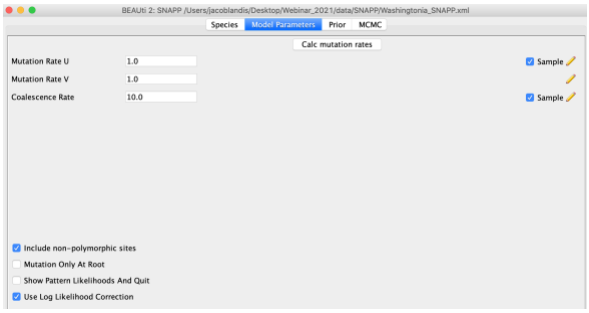
*vcftools --vcf Samples.vcf --keep SNAPP.txt --max-missing 1.0 --thin 5000 --recode --recode-INFO-all --out Samples_SNAPP*

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

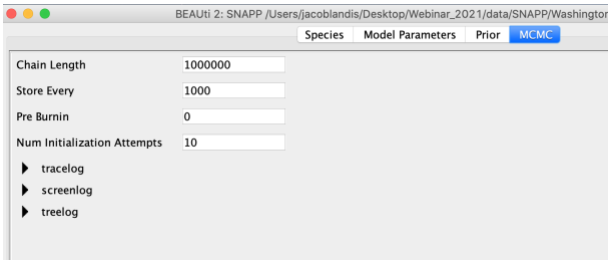
**Also important, make sure to use the newest version of Beast 2.6.3. Older versions look for different versions of the dependencies, and things will fail. [https://www.beast2.org/](https://www.beast2.org/)**

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”

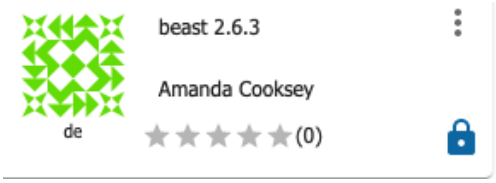





MCMC tab, set to 100,000 generations as a test (full runs will take much longer)
Save XML file.

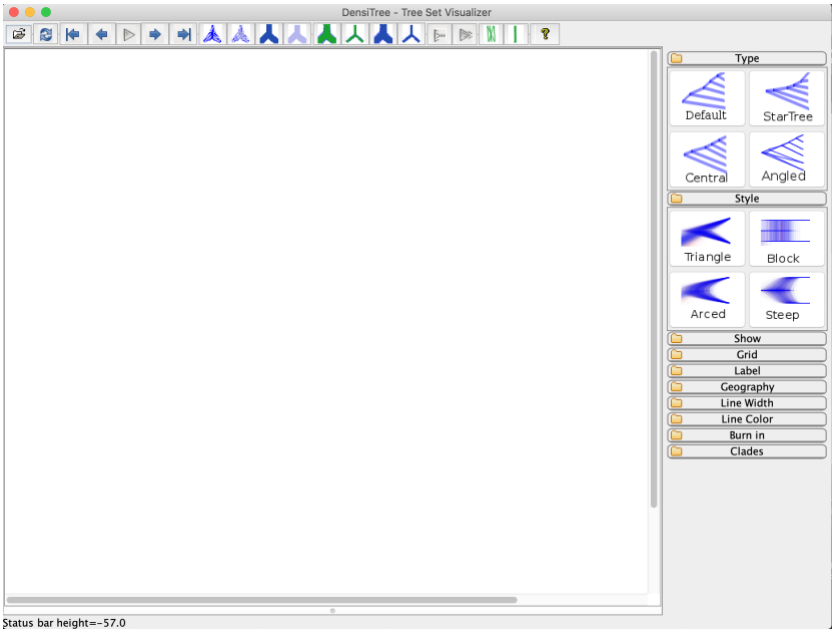



Use the Beast 2.6.3 app in DE

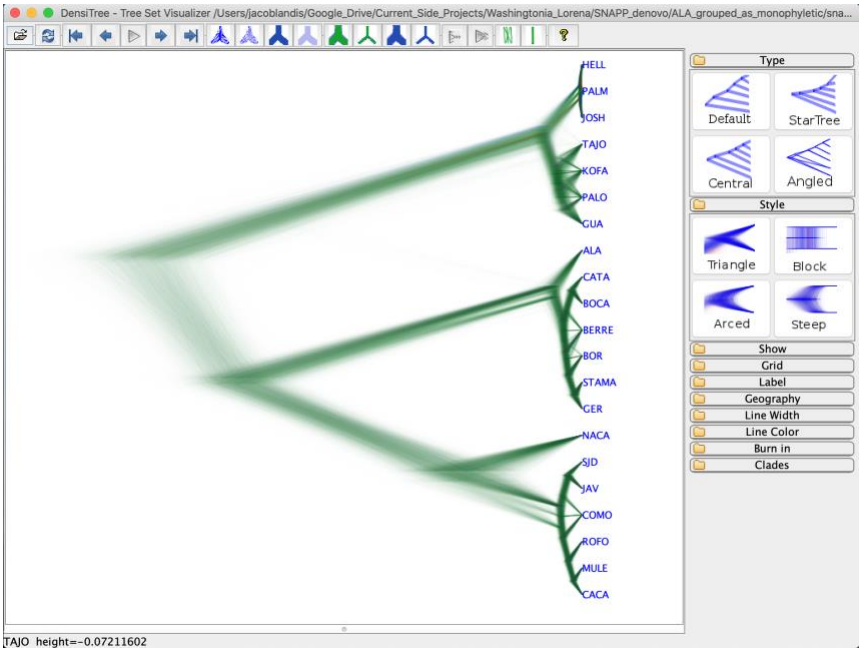



Use the generated XML file as the input (“Washingtonia_SNAPP_2.6.3.xml”), hit *Launch Analysis*.

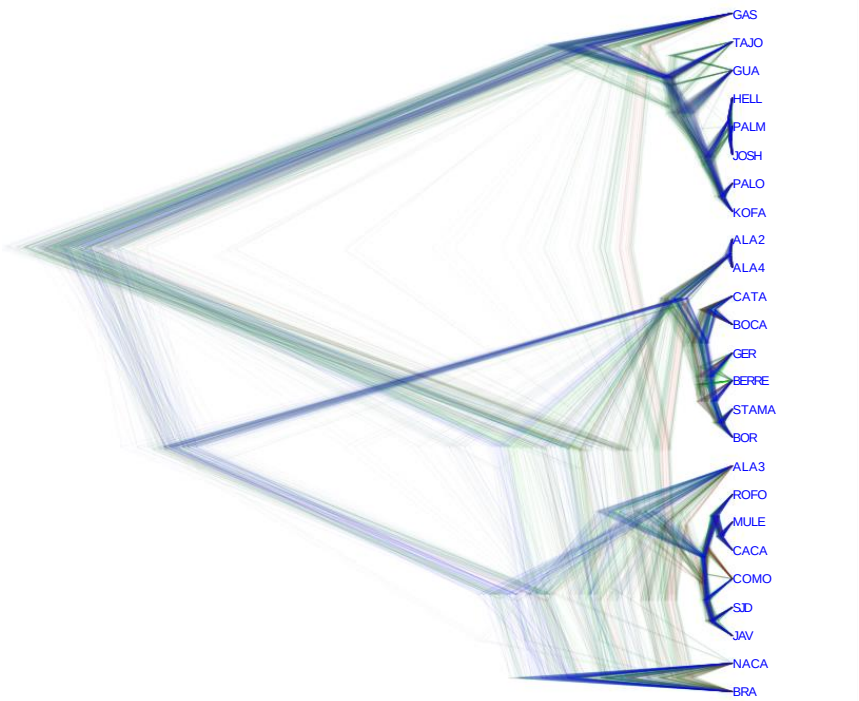
After the run completes, you can open the resulting snapp.trees file with DensiTree on your local machine.




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.

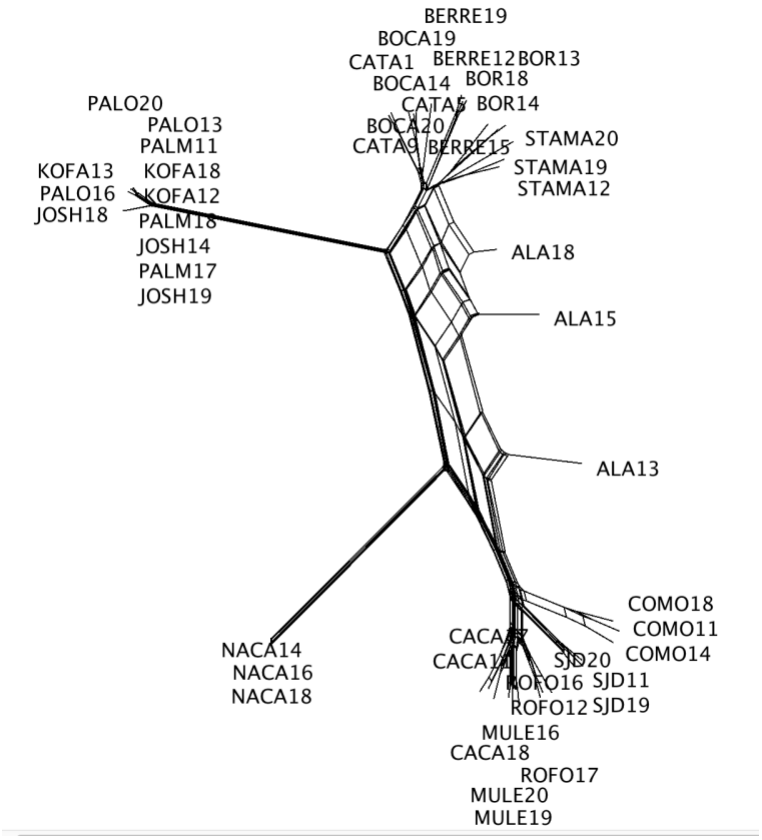
Download the appropriate version for your local machine

- 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.

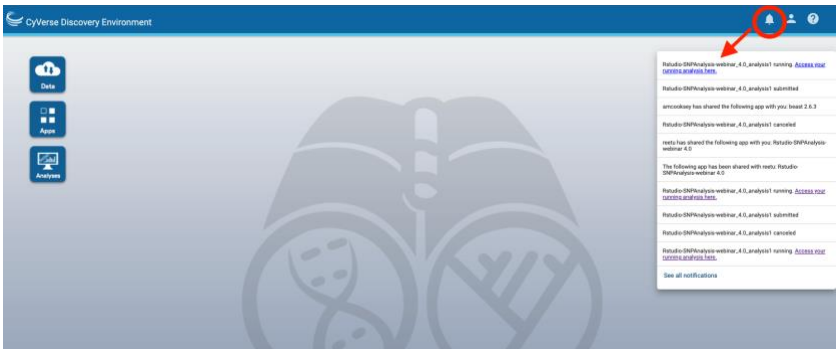


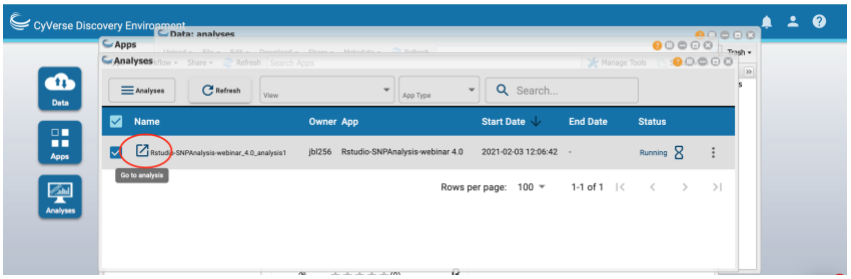

## PCA (Rstudio-SNPAnalysis-webinar)


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.

First up, launch a Rstudio instance by selecting the folder location of the necessary files.
*Input data* - select folder where your files are located for this analysis, in this case PCA
*Launch analysis*

This may take a few moments to full launch in VICE, but when it is ready you will be able to open a web browser with a new Rstudio instance
To launch, click on the icon just to the left of the instance name after clicking on your notifications icon and selecting the Rstudio analysis:




Once your Rstudio window is open, first set your working directory:

Session -> Set Working Directory -> Choose Directory. Select the folder you selected before for launching the App, in this case it is called PCA.

You should have the following files in the folder:

- Data file in VCF format: Washingtonia_test_samples.recode.p.snps.vcf
- Main Rscript with all commands: SNPRelate_CyVerse.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

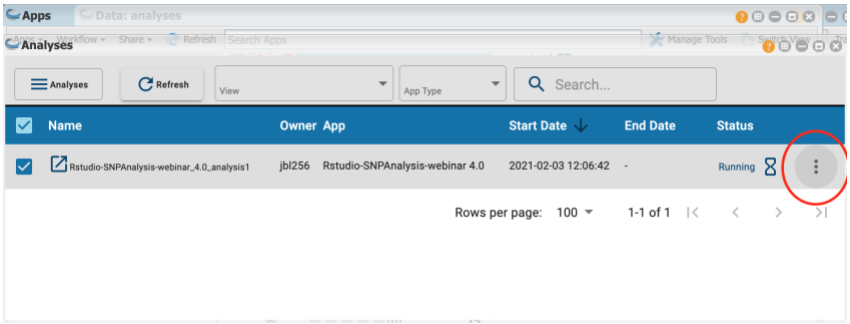
Then open the Rscript that contains all the code for generating and saving the plots.
File -> Open File -> SNPRelate.R

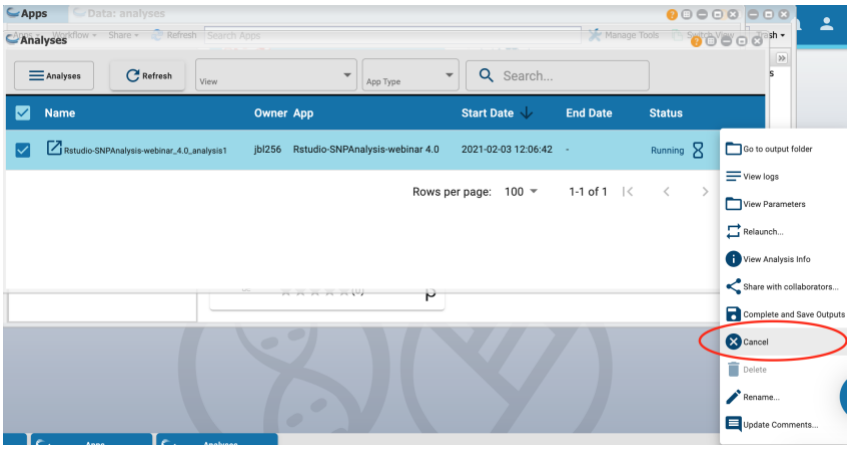
Run through the R script as you would on your local machine.

The script is set up to save a PDF of each plot in the home directory of the virtual machine with the Rstudio instance. To view these, click File -> Open File, and select the one wanted. This will open a new tab, in which you can easily download to your local machine. Important to note, once you terminate the Rstudio instance (see below), any files on the virtual machine will be lost. Make sure to transfer/save any files that you want to keep.

Default settings in the Discovery Environment is to only allow 2 instances of Rstudio at a time. So to be efficient with resources and to not limit the launch of other jobs, you need to

cancel/close the Rstudio instance when you are done with this particular analysis. To end the instance click on the three little dots next to the analysis and click cancel:




## Structure (Rstudio-SNPAnalysis-webinar)


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.

Similar to the PCA you just did, you will need to launch a new Rstudio instance but this time select the folder with the Structure data (if in a different location).

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_Cyverse.R script with all the commands

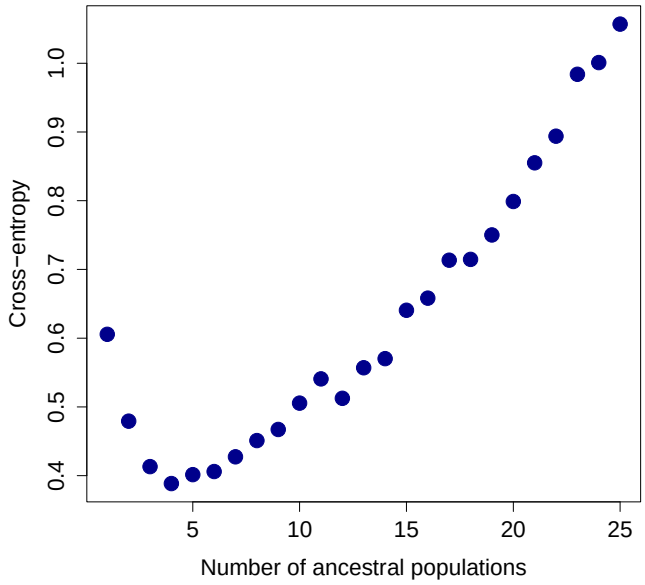
- Two additional R files: POPSutilities.R and Conversion.R, which were downloaded from the LEA package

Open the Rscript that contains all the code for generating and saving the plots.
File -> Open File -> Structure_Cyverse.R

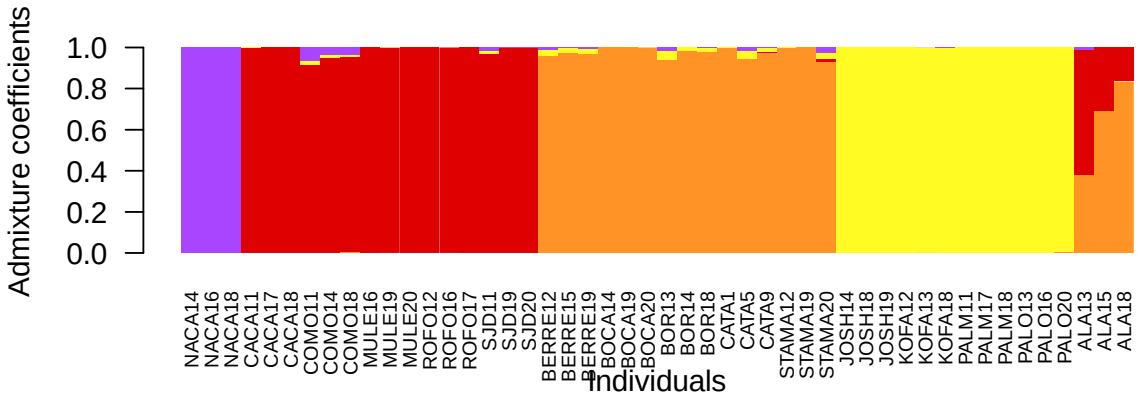
Run through the R script as you would on your local machine.

The script is set up to save a PDF of each plot in the home directory of the virtual machine with the Rstudio instance. To view these, click File -> Open File, and select the one wanted. This will open a new tab, in which you can easily download to your local machine. Important to note, once you terminate the Rstudio instance (see below), any files on the virtual machine will be lost. Make sure to transfer/save any files that you want to keep.

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-SNPAnalysis-webinar)


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:

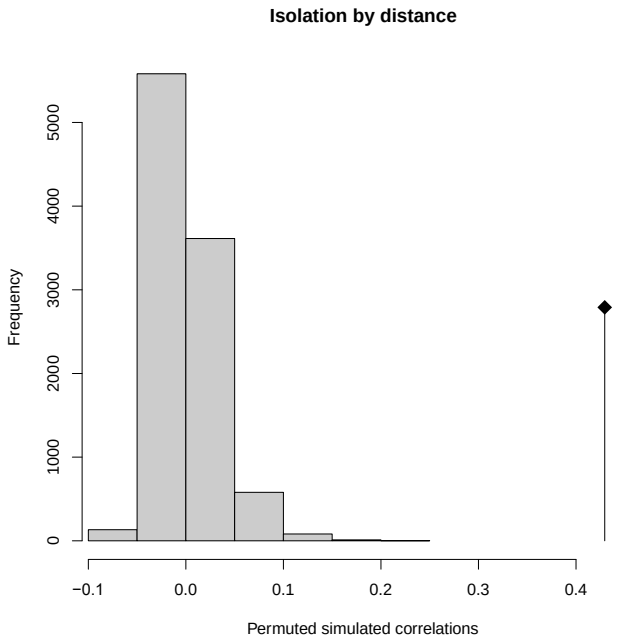
- Data file produced by Stacks populations (with a modified extension of .gen instead of .genpop): Washingtonia_test_samples.recode.p.snps.gen
- The Rscript with all the steps: IBD_Cyverse.R
- Latitude and Longitude values for each sample in decimal format: latlong.txt

Make sure to set the working directory to the selected folder. Open the Rscript that contains all the code for generating and saving the plots.
File -> Open File -> IBD_Cyverse.R

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.

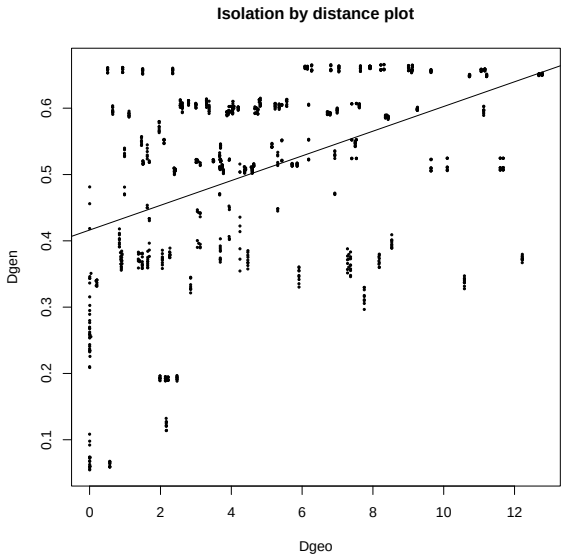
The script is set up to save a PDF of each plot in the home directory of the virtual machine with the Rstudio instance. To view these, click File -> Open File, and select the one wanted. This will open a new tab, in which you can easily download to your local machine. Important to note, once you terminate the Rstudio instance (see below), any files on the virtual machine will be lost. Make sure to transfer/save any files that you want to keep.

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

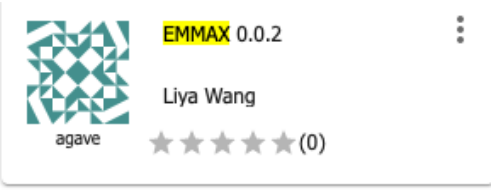
## EMMAX (two Apps needed; first Plink 1.904b and then EMMAX 0.0.2)


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.

First need to reformat our VCF file into Plink format using the Plink App.
*Input file* is our GWAS VCF file "Washingtonia_modified_GWAS_genotype_data.vcf"
*Command Line Extension:* We need to add additional command line flags that are not available in the standard Plink app. These can be found in the file "ext.txt". These include "--maf 0.05 --double-id --recode 12 --out plink_filter --output-missing-genotype 0 --transpose --allow-extra-chr".
*Launch Analysis*

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 two steps, estimating the kinship file and running the test for association will be done all at the same time. If you already have a kinship file, that can be specified as well.
*Inputs*:

Plink tped: plink_filter.tped
Trait file (from our original GWAS folder, only one trait at a time):
Plink tfam: plink_filter.tfam

*Parameters:*

Kinship estimation method: BN
Number of header lines in trait file: 0

(for our files there are not header rows, but if you are using your own data this should be modified).

*Launch analysis.*

This will create the kinship matrix (unfortunately it doesn't save the file). The run creates two files that can be used to plot the Manhattan plot: EMMAX.ps and manhattan.plot. We will use the file manhattan.plot since the Rscript for plotting is formatted to this specific file (though this will be a bit different if you are using EMMAX on your own Linux system). We also need the map file (plink_filter.map) from our Plink run which gives us the chromosome positions. Slight formatting needs to be done to make this ready for input for Rstudio though. To do this via the command line on your local machine (this file has already been produced and is in the EMMAX folder, the following command is necessary (once you are in the appropriate folder with the map file):

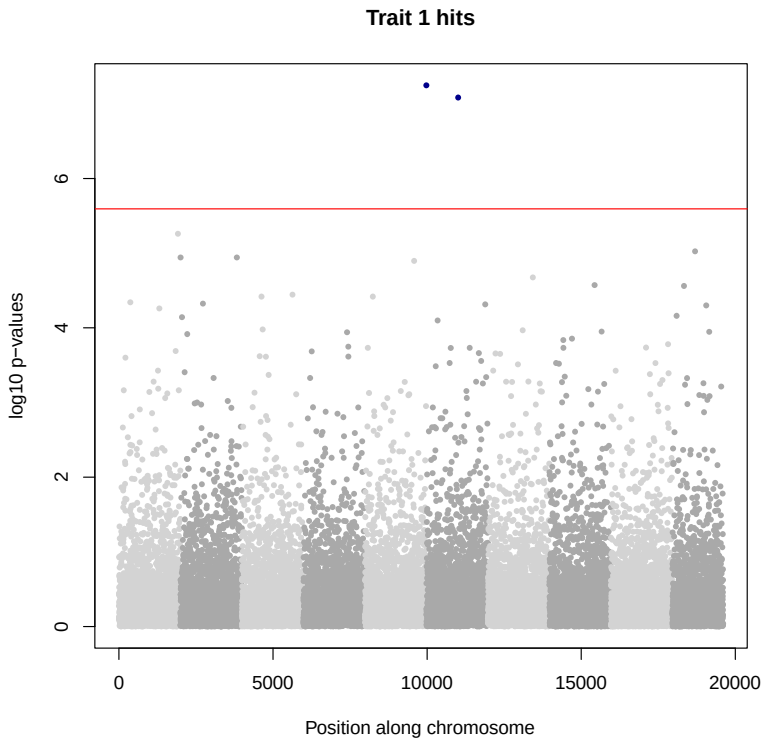
awk '{print $1,$2}' plink_filter.map > position.txt

To plot, start a new Rstudio instance on Cyverse.


Make sure to have the following files are in the appropriate folder and launch an Rstudio instance:

- The Rscript with all the commands: gwas_plot_CyVerse.R
- Results from the EMMAX run: manhattan.plot
- Position of chromosomes and base positions: position.txt

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