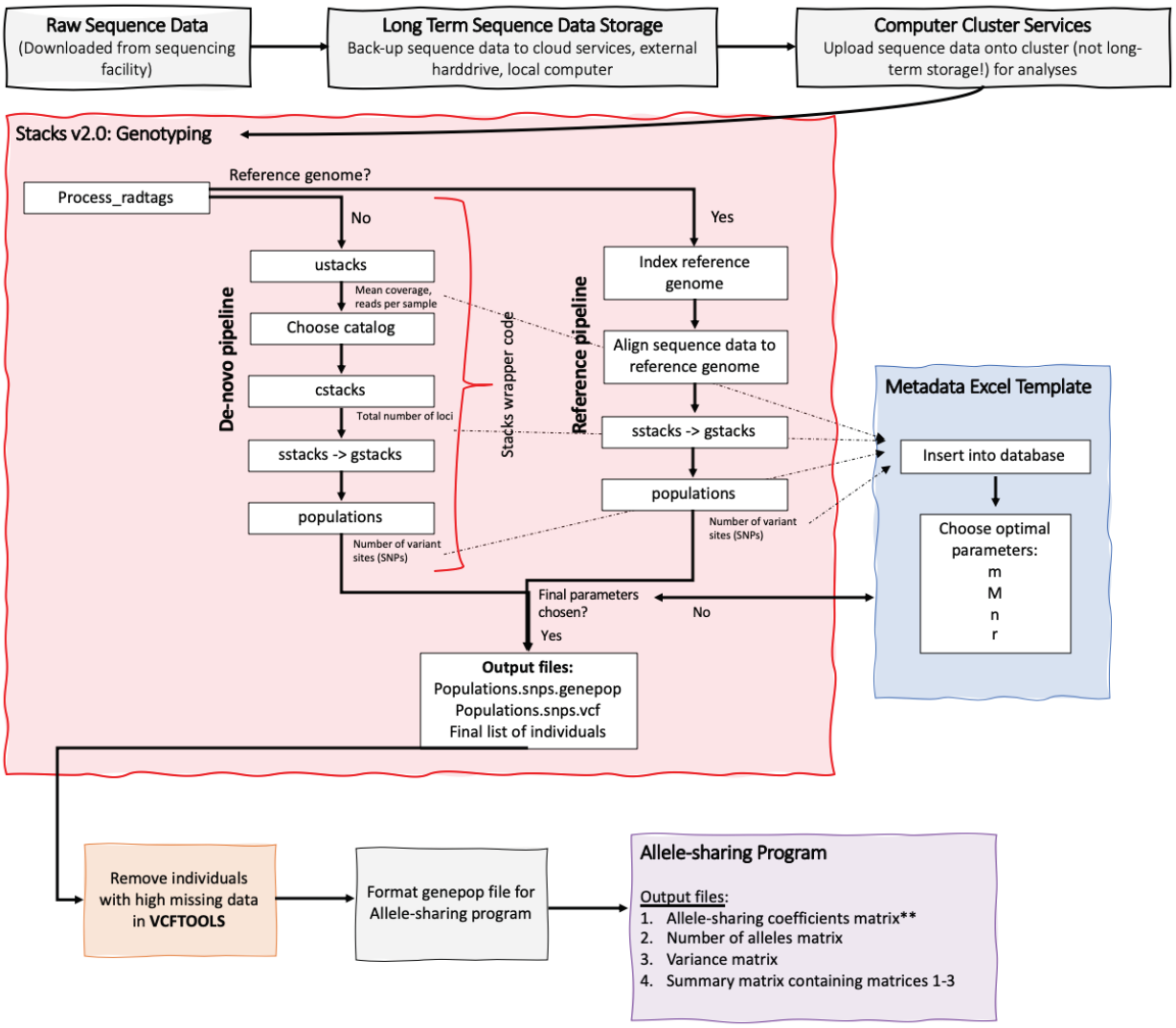
Standardized Data Analysis guide

A guide to the analysis of genomic data for improved management of AZA's breeding programs

Samantha Hauser, University of Wisconsin-Milwaukee
Jamie Ivy, San Diego Zoo Global
Andrea Putnam, San Diego Zoo Global
Emily Latch, University of Wisconsin-Milwaukee

Version 1.0. Published June 15, 2020

Standardized Data Analysis Workflow



Overview

The purpose of this guide is to provide a standardized protocol for analyzing your sequence data to the end result of allele-sharing values among all individuals in your analysis.

We provide information on a) best practices to store your sequence data, b) computing options for running these analyses, c) Stacks protocol using a provided wrapper code, d) an alternative manual Stacks protocol using provided template codes, e) filtering and formatting for the genotype data, and f) an allele-sharing program and protocol that calculates allele-sharing coefficients and supplemental information.

Table of Contents:

You Have Your Sequence Data – Now What?

Download your data and save onto multiple places (this might take a while – the files are large!):

a. Hard drive* (multiple may not be a bad idea)
b. Cloud services
c. Local computer

You will next need to figure out how are you going to run the following data analyses (local computer vs computer cluster)?

Often times, computers are unable to handle the size of the genomic data in terms of space and computing power, and you may need to use a computer cluster. If your institution has access to one or can get access to one via a cloud service (For example: Amazon Cloud Services), we recommend you use those services as they will dramatically cut down on computing time.

Though we can provide some general recommendations and guidelines (below), computer clusters can vary in terms of access, use of a terminal, language for submitting jobs, etc. and we highly recommend talking to the cluster provider for more information and guidance for aid.

Some general recommendations/helpful tips:

1. Use a Secure File Transfer Program (SFTP) such as Cyberduck (Mac) or PuTTy (PC) to move files, rename files/folders, etc. between your local computer and the cluster and within the cluster.
2. We do recommend use of a Mac if possible as the text edit programs are inherently unix-based and will prevent downstream formatting errors in your files.
    a. If you do use a PC, make sure to use a text edit program either on the cluster (such as Nano) or on your local disk that you can ensure will be unix-friendly.
3. Make sure to have a notebook and record all your steps along the way (similar to the way you would keep a lab notebook).
4. Make sure you back up your data, analyses files, etc. as you analyze.

Once you have your sequence data backed up and in the location for your analyses, continue with:

Running Stacks with our wrapper code on page 5
Running Stacks manually on page 6

Running Stacks using Our Wrapper Code

What Stacks does: Stacks is a software pipeline for identifying SNPs, building loci from short-read sequencing, such as RAD sequencing, and calling genotypes for the purpose of population genomics. This pipeline will take your raw genomics sequences and output datasets consisting of SNP genotypes for each individual.

In this pipeline, you will need to optimize parameters in order to obtain as many informative genetic markers as possible, including the following parameters:

- m = minimum stack depth (default =3)
- M = max distance between stacks (default = 2)
- n = number mismatches allowed between loci and catalog (default = 1)
- r = min % of individuals (per pop) at which a locus must be present

To keep track of the meta-data associated with this optimization and eventually choose your optimal parameter settings, we recommend setting up an Excel workbook (hereafter metadata database) as follows. *We also provide a template Excel workbook in the associated files.*

m	M	ustacks output Mean Coverage	ustacks output Number of Reads	n	cstacks output Number of Loci	populations r 0.80 Number of Loci	populations r 0.60 Number of Loci	populations r 0.40 Number of Loci
1	2							
2	2							
3	2							
3	2							
3	2							
3	2							
3	2							
3	1							
3	3							

Our wrapper code (attached) runs through the Stacks bioinformatic pipeline for you and provides the necessary output files. You will need to go through the output files and record the metadata to choose the optimized parameter settings regardless.

NOTE: You may not be able to use the wrapper code with your computer or computing cluster. If you cannot or choose not to use the wrapper for another reason, see the next section: "Running Stacks Manually" on page 6.

Once you have run the wrapper code, go to page 18 on for the next section: "Choosing Your Optimal Parameters"

Running Stacks Manually (Stacks version 2.0)

If for whatever reason you cannot or choose not use the Stacks wrapper code (above), you can run Stacks manually. The following section is to guide you how to run the program step-by-step to mimic the wrapper for double digest RAD (ddRAD) sequencing.

We provide all the sample code as text documents which you can modify for your project.

Please reference the Stacks manual for more help: http://catchenlab.life.illinois.edu/stacks/manual/

And/or the Stacks online forum for questions: https://groups.google.com/forum/#!forum/stacks-users

---

What Stacks does: Stacks is a software pipeline for identifying SNPs, building loci from short-read sequencing, such as RAD sequencing, and calling genotypes for the purpose of population genomics. This pipeline will take your raw genomics sequences and output datasets consisting of SNP genotypes for each individual.

In this pipeline, you will need to optimize parameters in order to obtain as many informative genetic markers as possible, including the following parameters:

- m = minimum stack depth (default =3)
- M = max distance between stacks (default = 2)
- n = number mismatches allowed between loci and catalog (default = 1)
- r = min % of individuals (per pop) at which a locus must be present

To keep track of the meta-data associated with this optimization and eventually choose your optimal parameter settings, we recommend setting up an Excel workbook (hereafter meta-data database) as follows. *We also provide a template Excel workbook in the associated files.*

m	M	ustacks output Mean Coverage	ustacks output Number of Reads	n	cstacks output Number of Loci	populations r 0.80 Number of Loci	populations r 0.60 Number of Loci	populations r 0.40 Number of Loci
1	2							
2	2							
3	2							
3	2							
3	2							
3	2							
3	2							
3	1							
3	3							

Running Stacks Manually Protocol (Version 2.0)

1. **Process Radtags:** This process separates and sorts out your individuals' sequence data (called demultiplexing) and cleans and filters the data.

To demultiplex or not?

Without demultiplex: If your raw sequences are already sorted by individual (some genomics facilities will do this for you) – you still need to clean and filter your data so don't skip this step!

```
process_radtags -T 24 -P -p ./ --disable_rad_check -o ../cleaned
-c -q --renz_1 'speI' --renz_2 'sau3AI' -i gzfastq
```

-T = number of tasks
-P = paired
-p = directory with files
--disable_rad_check = don't check for rad sites
-c = clean
-q = quality filter
-renz_1 = specify enzyme #1
-renz_2 = specify enzyme # 2
-i = specify file type
-o = specify output folder

With demultiplex: If your raw sequence files come in 1-2 large zipped files then you will need to sort them by individual. You will need a barcode file - a text file that has the specific barcodes for each individual. The genomics facility should provide you with this information.

```
process_radtags  -T 24\
 -1 ./WLHMS_5/WLHMS_5_USPD16097349_HKFHKDSXX_L3_1.fq.gz \
 -2 ./WLHMS_5/WLHMS_5_USPD16097349_HKFHKDSXX_L3_2.fq.gz \
 -b ../info/HMS_Barcodes1.txt \
-c -q --renz_1 'speI' --renz_2 'sau3AI' -i gzfastq
```

-1 and -2 = says that it is paired and data are in these two files
\ = allows you to continue code on next line
-b = demultiplex with barcode file

The output of this program will be put into the folder named "cleaned" with 4 output files for each individual.

Are you running with reference genome or without?
With a reference genome: Go to page 15 to continue.
Without a reference genome: Continue with the De-novo pipeline below

De-Novo Pipeline

2. ustacks: this step creates stacks of SNPs found in the sequences that you will then filter through and choose the best ones in later steps in this pipeline.
You will need to make a population map (a text file that shows what population each individual is in). For most captive populations, all individuals are managed as a single population so all individuals will have the same population name (here, "Zoo").

Example population map (in a text file):

```
Fox_673 Zoo
Fox_700 Zoo
Fox_701 Zoo
Fox_737 Zoo
Fox_785 Zoo
Fox_799 Zoo
Fox_824 Zoo
Fox_833 Zoo
```

You will be optimizing parameters m and M in this step by running the following combinations in ustacks.

m	M
1	2
2	2
3	2
3	1
3	3

NOTE: You will have extra m = 3, M = 2 rows in your metadata database. You will need those extra rows in a later step. For this step, enter the same information for all rows.

```
mkdir ../stacks.denovo/m1M3

j=1
prefix="./cleaned"
output="./stacks.denovo/m1M3"



for f in $(awk '{print $1}' < FennecFox.popmap.txt); do
    ustacks -t gzfastq  \
            -f "$prefix$f.fq.gz" \
            -o "$output"   \
            -i $j \
            -m 1 -M 3 -p 24   \
            --model_type snp \

    j=$(($j + 1))
done
```

You will need to modify the example code with your directory file (for each combination you are running (m1M2, m3M2, etc.), your population map file name, and update the -m and -M values to match the combination you are running.

The output files (3 per individual) will be automatically put into the directory file you assigned. There will also be a log file created that includes information on the coverage and number of reads for each individual:

```
Final coverage: mean=4.04; stdev=3.28; max=78; n_reads=27442(26.5%)
```

Copy this information into an Excel file, so that you have mean coverage (mean=) and number of reads (n_reads = [without the percentage]) in a separate sheet in your metadata Excel file, as follows. You will need to do this for each m and M combination.

	ustacks	
Sample	Mean_Cov	N_reads
1012	10.42	1174982
1013_D	8.41	878509
1014_D	4.98	720499
1005_D	8.74	915815
876_D	9.58	1043676
962_D	8.37	777926
963	8.77	911029
1031	11.17	1689525
1013_D2	7.63	2074902
1021	10.25	1171489
993_D	10.94	1295272
933_D	5.99	1046331
1121_D	10.18	1215690
1120_D	8.75	922898
1122_D	10.19	1168649
1014_D2	7.18	1742567
673_D	10.17	1137188

You will also need to summarize the coverage for each parameter (average) into your metadata database.

For each parameter combination (e.g., m = 2, M = 2), using the Excel function =AVERAGE:

Mean Coverage =	16.0946341
Mean N Reads =	3042974.73

For all parameter combinations:

m	M	ustacks output Mean Coverage	ustacks output Number of Reads
1	2	3.46	1504262
2	2	16.09	3042975
3	2	21.52	3153210
3	2	21.52	3153210
3	2	21.52	3153210
3	2	21.52	3153210
3	2	21.52	3153210
3	1	21.03	3150165
3	3	21.78	3172302

3. cstacks: this step creates a catalog of SNPs from your best individuals (those with the highest quality sequence data).

To create your catalog, you must choose individuals from the top 25% based on their mean coverage values. To choose the particular individuals, use the coverage, number of reads information in the metadata sheet you collected in the last step.

	ustacks	
Sample	Mean_Cov	N_reads
1012	10.42	1174982
1013_D	8.41	878509
1014_D	4.98	720499
1005_D	8.74	915815
876_D	9.58	1043676
962_D	8.37	777926
963	8.77	911029
1031	11.17	1689525
1013_D2	7.63	2074902
1021	10.25	1171489
993_D	10.94	1295272
933_D	5.99	1046331
1121_D	10.18	1215690
1120_D	8.75	922898
1122_D	10.19	1168649
1014_D2	7.18	1742567
673_D	10.17	1137188

Your catalog text file will look like:

```
2870    Zoo
2871    Zoo
2872    Zoo
2909    Zoo
1011    Zoo
700     Zoo
701     Zoo
737     Zoo
785     Zoo
799     Zoo
833     Zoo
869     Zoo
870     Zoo
880     Zoo
898     Zoo
```

Optimizing n: For this optimization, you will only be running multiple n values for m3M2. The rest of the folder (m1M2, m2M2, etc.) you will only be running n=0. This is so that you can properly compare n values without other variables affecting the resulting metadata you will collect.

Optional: To save some time, you can copy folder m3M2 and create folders m3M2n0, m3M2n1, m3M2n2, m3M2n3, m3M2n4. This will allow you to run multiple n values for the m3M2 database for optimization.

```
cstacks -P ./m3M2 -M ../info/m3M2_catalog.txt -n 1 -p 24
```

- -P = input file directory
- -o = output directory
- -M = catalog
- -n = number of mismatches with catalog
- -p = number of tasks

All the combinations you will need to run (already listed in your metadata database):

m	M	n
1	2	0
2	2	0
3	2	0
3	2	1
3	2	2
3	2	3
3	2	4
3	1	0
3	3	0

Output data will be put into the directory you are working. A log file will also be created in which it summarizes the number of loci created. Note that information into your metadata Excel file as follows:

At the end of the log file:

```
Writing catalog in directory './m3M3/'.
Final catalog contains 661694 loci.
cstacks is done.
```

Your metadata database:

m	M	ustacks output Mean Coverage	ustacks output Number of Reads	n	cstacks output Number of Loci
1	2	3.46	1504262	0	534038
2	2	16.09	3042975	0	1492139
3	2	21.52	3153210	0	801534
3	2	21.52	3153210	1	734319
3	2	21.52	3153210	2	706521
3	2	21.52	3153210	3	689128
3	2	21.52	3153210	4	678682
3	1	21.03	3150165	0	813258
3	3	21.78	3172302	0	820056

4. sstacks → gstacks: In the next several steps, you are comparing each individual against the catalog in order to genotype each individual.
There is no optimization in these steps and therefore, you can run all these steps together in one code.

Run as follows for each combination thus far (see in cstacks).

```
sstacks -P ./m2M2 -M ../Fennec_popmap.txt -p 24

tsv2bam -P ./m2M2 -M ../Fennec_popmap.txt -R ../cleaned -t 24

gstacks -P ./m2M2 -M ../Fennec_popmap.txt -t 24
```

You will need to modify the example code for the appropriate folder (e.g., combination you are running) and your population map file name.

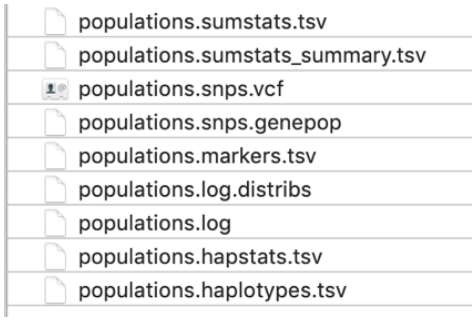
5. populations: creates the resulting genotype file for all individuals.

Optimizing r:
For all combinations listed in step 4 (cstacks) you need to run populations with the values 0.8, 0.6, and 0.4 (must run separately, and create new folders (e.g., pop 0.6) to move files into, otherwise you will overwrite the outputs)

```
populations  -P ./m3M3 -M ../Fennec_popmap.txt --write_single_snp
random --genepop --vcf -r 0.80 --min_maf 0.02
```

- -P = input file directory
- -M = popmap (NOTE different than other –M)
- --write_single_snp (can choose first or random) = choose 1 snp in a stack
- --genepop –vcf = output formats

Want to make sure that you output a genepop file and a vcf file! Note that the outputs will be generic – you will need to rename manually to be descriptive.



There will also be a populations.log file that you will need to access for your metadata information. The number of variant sites, this is at the end of the log file) is your number of SNPs:

```
Population summary statistics (more detail in populations.sumstats_summary.tsv):
  Zoo: 39.305 samples per locus; pi: 0.23435; all/variant/polymorphic sites: 21384642/25315/25315; private alleles: 0
Populations is done.
```

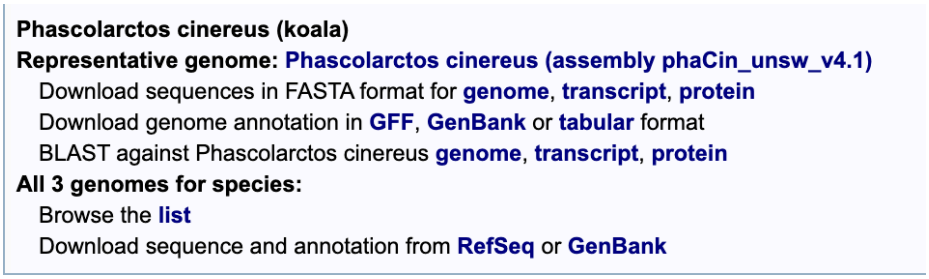
In your metadata Excel file, copy the final number of SNPs.

m	M	ustacks output Mean Coverage	ustacks output Number of Reads	n	cstacks output Number of Loci	populations r 0.80 Number of Loci	populations r 0.60 Number of Loci	populations r 0.40 Number of Loci
1	2	3.46	1504262	0	534038	0	150	500343
2	2	16.09	3042975	0	1492139	0	129	26128
3	2	21.52	3153210	0	801534	0	24	19598
3	2	21.52	3153210	1	734319	2	24	20384
3	2	21.52	3153210	2	706521	3	42	20637
3	2	21.52	3153210	3	689128	2	43	20827
3	2	21.52	3153210	4	678682	2	49	20896
3	1	21.03	3150165	0	813258	0	27	19252
3	3	21.78	3172302	0	820056	0	25	19544

From here go to page 18 to choosing your optimal parameters.

Reference Genome Pipeline

1. Reference Genome: If you know that a reference genome exists for your species and need to obtain the actual sequence data, you can visit the following page to search for it and download it: https://www.ncbi.nlm.nih.gov/genome/gdv/ or https://www.ncbi.nlm.nih.gov/genome/

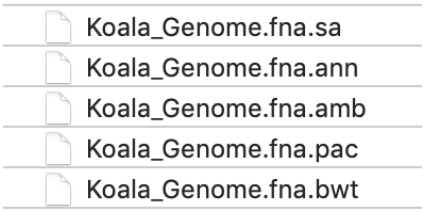


You will click on "Download sequence in FASTA format for genome" to download the actual sequence data necessary.

2. Index Reference Genome using software Bowtie with the following code:

```
bwa index Koala_Genome.fna  [change the file type manually from .fasta to .fna]
bwa aln
bwa sampe
```

You will have 5 output files from this process which you will need in order to align your sequence data to the reference genome:



3. Align Sequence Data to Reference Genome: You should have already run process_radtags with your sequence data. If you haven't, go back to page 7 and then come back to this step.

Align output files from process_radtags to reference genome using the following code for an individual:

```
bwa mem -M ../genome/Koala_Genome.fna ../cleaned/Koala_123.1.fq.gz
../cleaned/Koala_123.2.fq.gz |
        samtools view -h -b -S|
        samtools sort > ./alignments.bwa/Koala_123.bam
```

4. sstacks → gstacks: In the next several steps, you are comparing each individual against the catalog in order to genotype each individual.
There is no optimization in these steps and therefore, you can run all these steps together in one code.
Run as follows for each combination thus far (see in cstacks).

```
sstacks -P ./m2M2 -M ../Fennec_popmap.txt -p 24

tsv2bam -P ./m2M2 -M ../Fennec_popmap.txt -R ../cleaned -t 24

gstacks -P ./m2M2 -M ../Fennec_popmap.txt -t 24
```

You will need to modify the example code for the appropriate folder (e.g., combination you are running) and your population map file name.

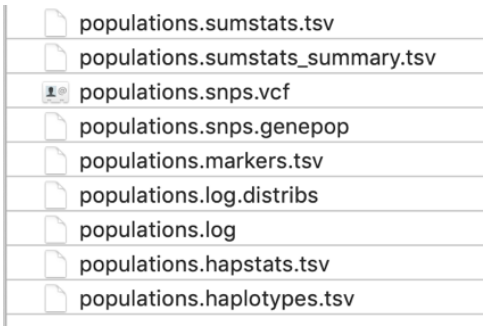
5. populations: creates the resulting genotype file for all individuals.

For all combinations listed in step 4 (cstacks) you need to run populations to optimize parameter -r : 0.8, 0.6, and 0.4 (must run separately, and create new folders (e.g., pop 0.6) to move files into, otherwise you will overwrite the outputs)

```
populations  -P ./m3M3 -M ../Fennec_popmap.txt --write_single_snp
random --genepop --vcf -r 0.80 --min_maf 0.02
```

- -P = input file directory
- -M = popmap (NOTE different than other –M)
- --write_single_snp (can choose first or random) = choose 1 snp in a stack
- --genepop –vcf = output formats

You want to make sure that you output a genepop file and a vcf file! Note that the outputs will be generic – you will need to rename manually to be descriptive.



There will also be a populations.log file that you will need to access for your metadata information. The number of variant sites is your number of SNPs:

```
Population summary statistics (more detail in populations.sumstats_summary.tsv):
  Zoo: 39.305 samples per locus; pi: 0.23435; all/variant/polymorphic sites: 21384642/25315/25315; private alleles: 0
Populations is done.
```

		ustacks output	ustacks output Number		cstacks output	populations r 0.80	populations r 0.60	populations r 0.40
m	M	Mean Coverage	of Reads	n	Number of Loci	Number of Loci	Number of Loci	Number of Loci
1	2	3.46	1504262	0	534038	0	150	500343
2	2	16.09	3042975	0	1492139	0	129	26128
3	2	21.52	3153210	0	801534	0	24	19598
3	2	21.52	3153210	1	734319	2	24	20384
3	2	21.52	3153210	2	706521	3	42	20637
3	2	21.52	3153210	3	689128	2	43	20827
3	2	21.52	3153210	4	678682	2	49	20896
3	1	21.03	3150165	0	813258	0	27	19252
3	3	21.78	3172302	0	820056	0	25	19544

From here go to page 18 to choosing your optimal parameters.

Choosing Your Optimal Parameters

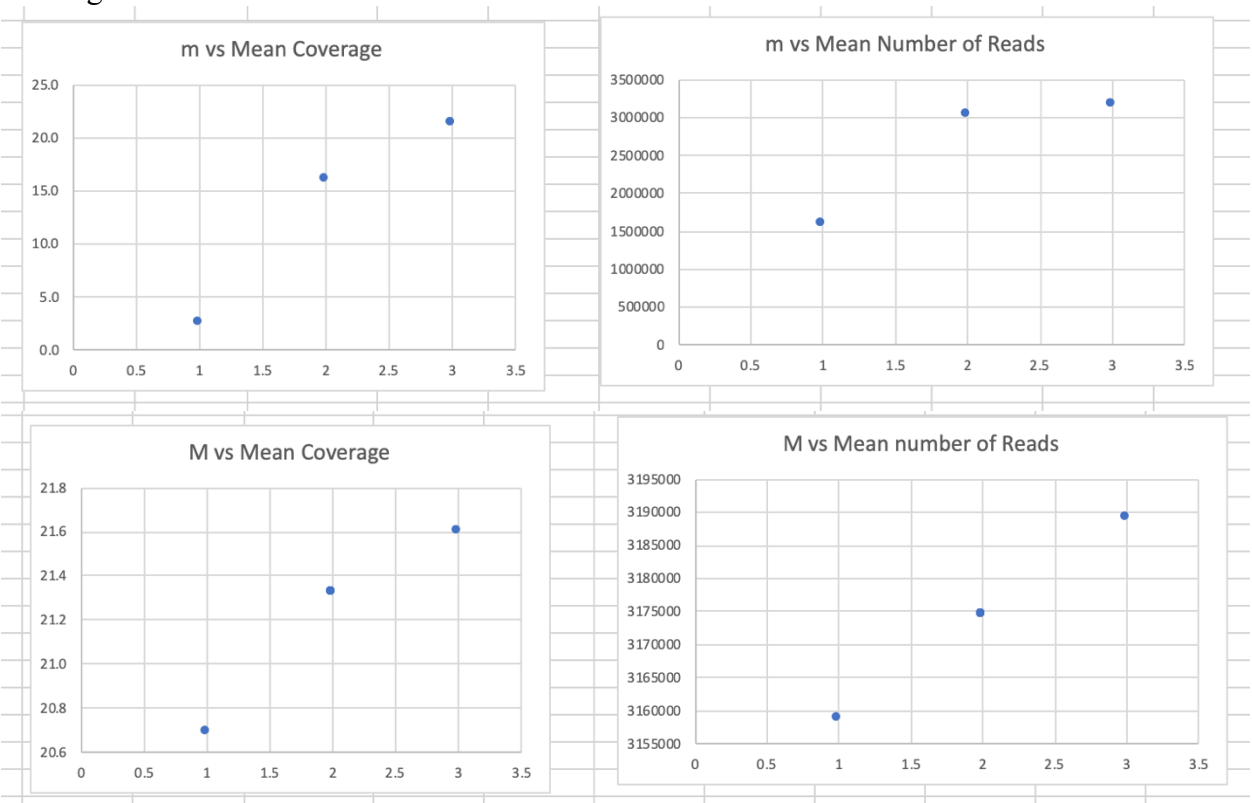
Now that you have run all the steps and optimization, you will need to determine what parameters are optimal.

You should have a metadata Excel file that is filled out as follows:

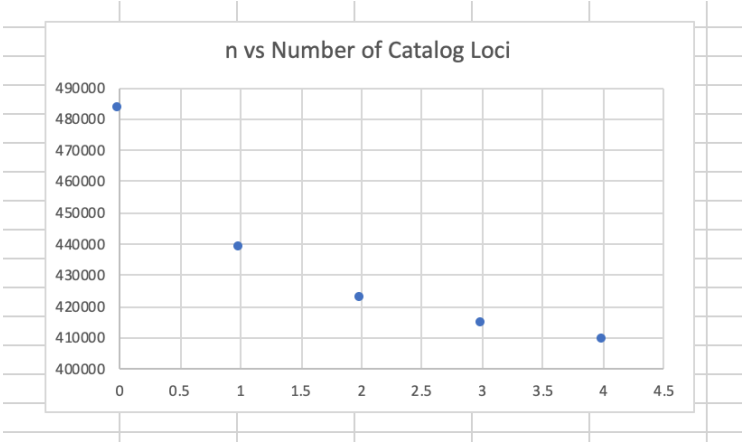
m	M	ustacks output Mean Coverage	ustacks output Number of Reads	n	cstacks output Number of Loci	populations r 0.80 Number of Loci	populations r 0.60 Number of Loci	populations r 0.40 Number of Loci
1	2	3.46	1504262	0	534038	0	150	500343
2	2	16.09	3042975	0	1492139	0	129	26128
3	2	21.52	3153210	0	801534	0	24	19598
3	2	21.52	3153210	1	734319	2	24	20384
3	2	21.52	3153210	2	706521	3	42	20637
3	2	21.52	3153210	3	689128	2	43	20827
3	2	21.52	3153210	4	678682	2	49	20896
3	1	21.03	3150165	0	813258	0	27	19252
3	3	21.78	3172302	0	820056	0	25	19544

To choose the following parameters, you will be plotting the variable against the output information you have collected in your metadata database as follows.

<u>Choosing m and M:</u> you will choose the combination of m and M that increase your mean coverage and number of reads.



Choosing n: you will choose the value of n the N of loci in your catalog.



Choosing r: you will choose the value of r where your number of loci is the highest.

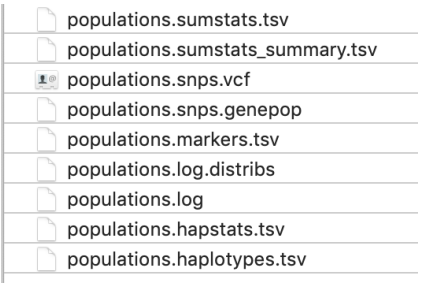
In this case, you would choose r = 0.40 as the other values give you very few SNPs.

populations r 0.80 Number of Loci	populations r 0.60 Number of Loci	populations r 0.40 Number of Loci
0	150	500343
0	129	26128
0	24	19598
2	24	20384
3	42	20637
2	43	20827
2	49	20896
0	27	19252
0	25	19544

Only if necessary: Run your optimal parameters through the program.
You may get a combination of optimal parameters that you have not run (For example, m = 3, M= 3 , n =0, r =0.6). In that case, you will need to rerun the pipeline to match your optimal parameters. Follow the steps above and modify your code accordingly.

Output out of Stacks:
You should have 3 output files that you will use from here on out:

a) Genepop file
b) VCF file: you will use this in the next step for filtering.
c) A list of individuals in your analyses (this could be your population map, but you may remove individuals through your filtering)



Continue on the next page to format your files to calculate allele sharing between pairs of individuals.

Prepare Genepop File for Allele-Sharing Program

I. Remove individuals with high missing data

Input file: The 'populations.snps.vcf' file from populations. You can rename this file manually to something more descriptive such as "FennecFox.vcf"

To remove individuals with high missing data, which are not informative or useful, we will use the program VCFTOOLS. It allows you to filter through large genotype datasets easily. This would be impossible to do manually in Excel or a text edit program.

Code to use:

```
vcftools --vcf populations.snps.vcf --missing-indv
```

This code will output a text file with proportion of missing data for each individual. Copy and paste into Excel:

INDV	N_DATA	N_GENO	N_MISS	F_MISS
338	38356	0	38329	0.999296
433	38356	0	38333	0.9994
460	38356	0	2067	0.0538899
536	38356	0	1428	0.0372302
552	38356	0	1110	0.0289394
572	38356	0	38335	0.999452
573	38356	0	1334	0.0347794
591	38356	0	38353	0.999922
595	38356	0	38350	0.999844
698	38356	0	38349	0.999817
699	38356	0	38331	0.999348
700	38356	0	38354	0.999948
724	38356	0	28914	0.753833
725	38356	0	38329	0.999296
726	38356	0	1001	0.0260976

Sort the missing data column:

INDV	N_DATA	N_GENO	N_MISS	F_MISS
338	38356	0	38329	0.999296
433	38356	0	38333	0.9994
572	38356	0	38335	0.999452
591	38356	0	38353	0.999922
595	38356	0	38350	0.999844
698	38356	0	38349	0.999817
699	38356	0	38331	0.999348
700	38356	0	38354	0.999948
724	38356	0	28914	0.753833
725	38356	0	38329	0.999296
778	38356	0	38352	0.999896
779	38356	0	38353	0.999922
796	38356	0	38333	0.9994
800	38356	0	38328	0.99927

Choose a cut-off for missing data (ideally: 20% missing data, realistic example: 50% missing data, a high cut-off would be upwards of 75% missing data).

Remove those individuals with missing data that exceeds your threshold from the genepop file. You can do this manually in a text-edit program (such as TextWrangler) fairly easily by removing the entire row (i.e., individual) from the genepop file.

```
1018,   0404    0303    0103    0202    0303    0204
1052,   0204    0303    0101    0202    0303    0204
1058,   0204    0103    0101    0202    0000    0204
1103,   0204    0303    0101    0000    0303    0404
1119,   0204    0103    0101    0202    0303    0204
1124,   0204    0303    0101    0202    0303    0204
```

If you have any duplicate individuals, you will also want to remove the duplicate with the higher missing data value. Remove these the same way.

II. Remove header from genepop file

The allele-sharing program cannot take a genepop file with a header, so you will need to manually remove this header the same way you removed individuals with high missing data.

```
# Stacks v2.2; GenePop v4.1.3; September 05, 2019
2_73,3_313,7_33,8_29,12_210,17_118,21_216,24_98,29_84,32_21,33_56
pop
Fox_1003,   0000    0000    0000    0000    0000    0000    0000
```

Final genepop file should look like this (without a header):

```
Fox_1003,   0000    0000    0000    0000    0000
Fox_1005,   0000    0000    0000    0000    0000
Fox_1011,   0202    0404    0202    0303    0202
Fox_1012,   0000    0000    0000    0000    0000
Fox_1013,   0000    0000    0000    0000    0000
Fox_1014,   0000    0000    0000    0000    0000
Fox_1027,   0202    0000    0202    0303    0202
Fox_1031,   0202    0204    0203    0203    0202
Fox_1065,   0202    0000    0202    0303    0204
Fox_1106,   0000    0000    0000    0000    0000
```

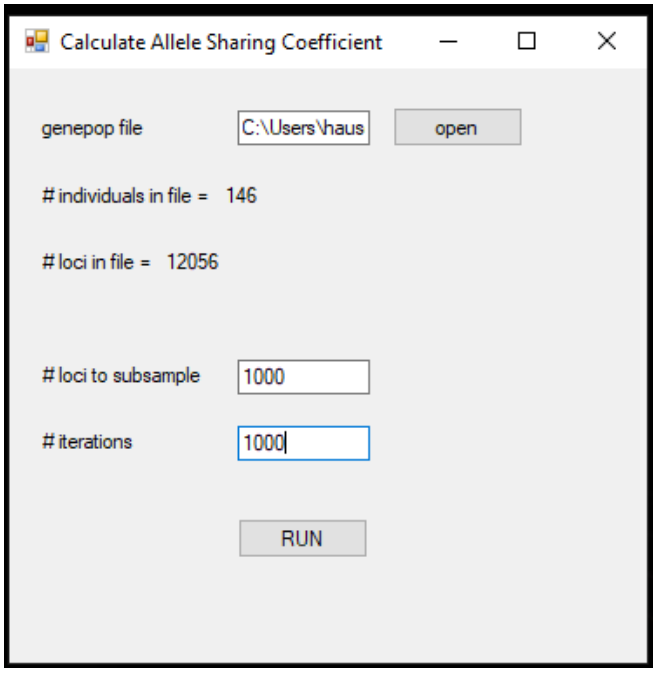
Allele-Sharing Program (CASC)

Input file: Prepared genepop file
If you have not prepared your genepop file, go back to page 20. It should look like this:

```
Fox_1003,   0000   0000   0000   0000   0000
Fox_1005,   0000   0000   0000   0000   0000
Fox_1011,   0202   0404   0202   0303   0202
Fox_1012,   0000   0000   0000   0000   0000
Fox_1013,   0000   0000   0000   0000   0000
Fox_1014,   0000   0000   0000   0000   0000
Fox_1027,   0202   0000   0202   0303   0202
Fox_1031,   0202   0204   0203   0203   0202
Fox_1065,   0202   0000   0202   0303   0204
Fox_1106,   0000   0000   0000   0000   0000
```

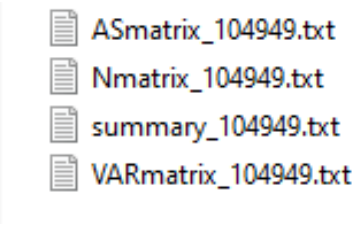
Steps

1. Upload your genepop file
2. Take note of the information populated by the genepop upload
   - Number of individuals
   - Number of loci
3. Input:
   - # loci to subsample = 1000
   - # iterations = 1000



Output files:

Note that there will be 4 output text files with generic names (as seen below). Be sure to save these with descriptive names so that you can refer back to them later.



Allele-sharing matrix (ASmatrix): contains the pairwise allele-sharing coefficient matrices that you will use in the following steps.

```
107,0.872832,0.746094,0.741331,0.748437,0.747374,
151,0.746094,0.876488,0.777660,0.736699,0.734138,
152,0.741331,0.777660,0.864777,0.724816,0.731298,
154,0.748437,0.736699,0.724816,0.859165,0.765260,
156,0.747374,0.734138,0.731298,0.765260,0.862996,
183,0.737902,0.790339,0.766378,0.735077,0.735678,
184,0.742398,0.772499,0.775055,0.737431,0.739759,
186,0.746154,0.788716,0.770040,0.734706,0.737118,
187,0.742259,0.801873,0.777471,0.732632,0.734104,
194,0.735011,0.746266,0.740014,0.733434,0.739764,
2135,0.730844,0.730196,0.726511,0.722228,0.726719
215,0.756260,0.741100,0.734504,0.741392,0.737075,
216,0.764067,0.753293,0.740417,0.746504,0.749372,
217,0.744036,0.753132,0.749098,0.734847,0.736760,
218,0.738729,0.740140,0.731311,0.735627,0.738547,
```

Number of alleles matrix (Nmatrix): contains the raw number of alleles shared between two individuals. You will not use in further analyses, but can be important in evaluating if there are any problems with your dataset (ex: batch effects, removing poor samples, etc.)

```
107,16948,16812,16609,16730,16746,16560,
151,16812,17042,16704,16821,16839,16645,
152,16609,16704,16839,16616,16637,16454,
154,16730,16821,16616,16959,16761,16566,
156,16746,16839,16637,16761,16977,16587,
183,16560,16645,16454,16566,16587,16784,
184,16737,16828,16627,16741,16766,16570,
186,16743,16840,16639,16757,16778,16584,
187,16768,16864,16667,16780,16798,16611,
194,16755,16846,16646,16766,16784,16591,
2135,15747,15843,15641,15759,15776,15601
215,16773,16865,16662,16781,16795,16606,
216,16777,16868,16665,16786,16804,16621,
217,15776,15863,15685,15789,15801,15624,
218,15639,15728,15531,15644,15659,15482,
```

Variance matrix (VARmatrix): contains the variance associated with the allele-sharing coefficient values. Also will not be used in further analyses, but important in quality control and evaluation if there are any analytical issues.

```
107,0.000046,0.000087,0.000086,0.000082,0.000083,
151,0.000087,0.000042,0.000078,0.000085,0.000091,
152,0.000086,0.000078,0.000047,0.000080,0.000085,
154,0.000082,0.000085,0.000080,0.000053,0.000067,
156,0.000083,0.000091,0.000085,0.000067,0.000049,
183,0.000087,0.000071,0.000080,0.000082,0.000084,
184,0.000086,0.000070,0.000075,0.000077,0.000076,
186,0.000079,0.000074,0.000073,0.000083,0.000079,
187,0.000089,0.000063,0.000073,0.000085,0.000084,
194,0.000085,0.000085,0.000081,0.000082,0.000087,
2135,0.000082,0.000084,0.000087,0.000081,0.000091
215,0.000079,0.000085,0.000087,0.000080,0.000091,
216,0.000073,0.000075,0.000085,0.000076,0.000077,
217,0.000076,0.000081,0.000073,0.000089,0.000083,
218,0.000085,0.000082,0.000093,0.000083,0.000086,
```

Summary matrix (summary): contains pairwise allele-sharing coefficient values, number of alleles shared, and variance of allele-sharing coefficients. Also will not be used in further analyses, but important in quality control and evaluation if there are any analytical issues.

```
107,107,0.872832,0.000046,16948
107,151,0.746094,0.000087,16812
107,152,0.741331,0.000086,16609
107,154,0.748437,0.000082,16730
107,156,0.747374,0.000083,16746
107,183,0.737902,0.000087,16560
107,184,0.742398,0.000086,16737
107,186,0.746154,0.000079,16743
107,187,0.742259,0.000089,16768
107,194,0.735011,0.000085,16755
107,2135,0.730844,0.000082,15747
107,215,0.756260,0.000079,16773
107,216,0.764067,0.000073,16777
107,217,0.744036,0.000076,15776
107,218,0.738729,0.000085,15639
107,228,0.746411,0.000090,16673
107,229,0.741699,0.000082,16446
107,232,0.741342,0.000089,16614
107,234,0.755745,0.000082,16724
107,236,0.738195,0.000075,10509
107,237,0.743282,0.000085,16532
```

Congrats! You now have allele-sharing values!!
To now incorporate these allele-sharing values into your pedigree go to X.