POPEase

Version 1.0
Release May 4, 2016

Louisiana State University

Table of Contents

POPEease

POPEase can be located on github at: https://github.com/reige012/POPEase

POPEase can be run on both Windows and Mac platforms.

About POPEase

Program Purpose

Are you new to population genetics? Not familiar with command-line programs, or just want a simple way to run basic pop gen analyses right on your laptop? Then POPEase is perfect for you. The purpose of the POPEase program is to help novices obtain reliable pop gen analyses for SNP data in just a few simple steps.

What The Program Does

POPEase (and the associated software) completes analyses that are normally done using a series of GUI or command-line programs that can be complicated for beginners. These programs are integrated into the script and run with a few simple command-line flags detailed in the program documents. The programs integrated into this script are standard analyses when examining population genetics of your data set.

This single python script bypasses the need to understand parsing your file into the correct format for the downstream analyses, it bypasses FSAT and/or Genepop to provide population genetics statistics, and it bypasses all programs associated with population structure including Structure, CLUMPP or CLUMPPAK, distruct, and Structure Harvester.

Program Caveats

First and foremost, POPEase is written to be compatible with python version =>3.0. It will not work with python 2.7 and you will need to update your python to the current version. You can get the downloadable version of python here.

POPEase accepts a VCF file of DNA SNP data. The program is currently only able to handle VCF data for diploid individuals and is defaulted to output genotypes as missing if the read depth is not >5. EasyPOP also excludes non-polymorphic SNPs. Phred Quality scores for VCF input are defaulted to only be accepted if they are >20 (99% accurate base call) or better. However, if desired, some of these caveats can be altered in the spid.spid file following the instructions in the documentation.

POPEase is written to be easy to use on your personal laptop or desktop. It can take a long time to run depending on the size of your data set, and may require the computer to be on and working for long periods of time. Please be aware of this when starting the program. You cannot pause it while Structure (the longest piece) is working.

Dependencies

All dependencies can be downloaded via Anaconda or from the respective websites for each program. Websites listed in Getting Started.

1. Python 3.0
2. Numpy
3. MatPlotLib
4. Pandas
5. Structure software
6. PGDSpider Software
7. Java

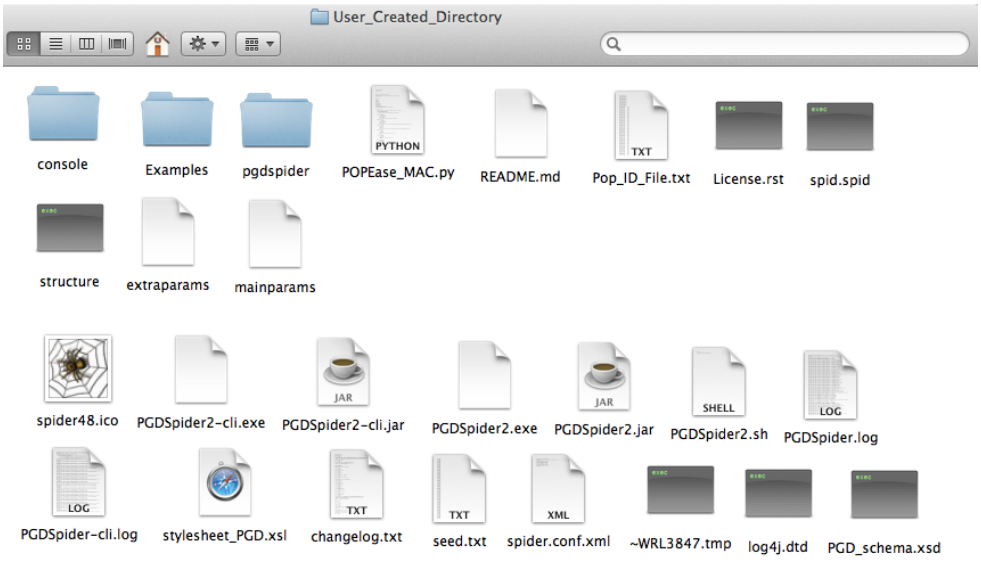
Instructions

Getting Started

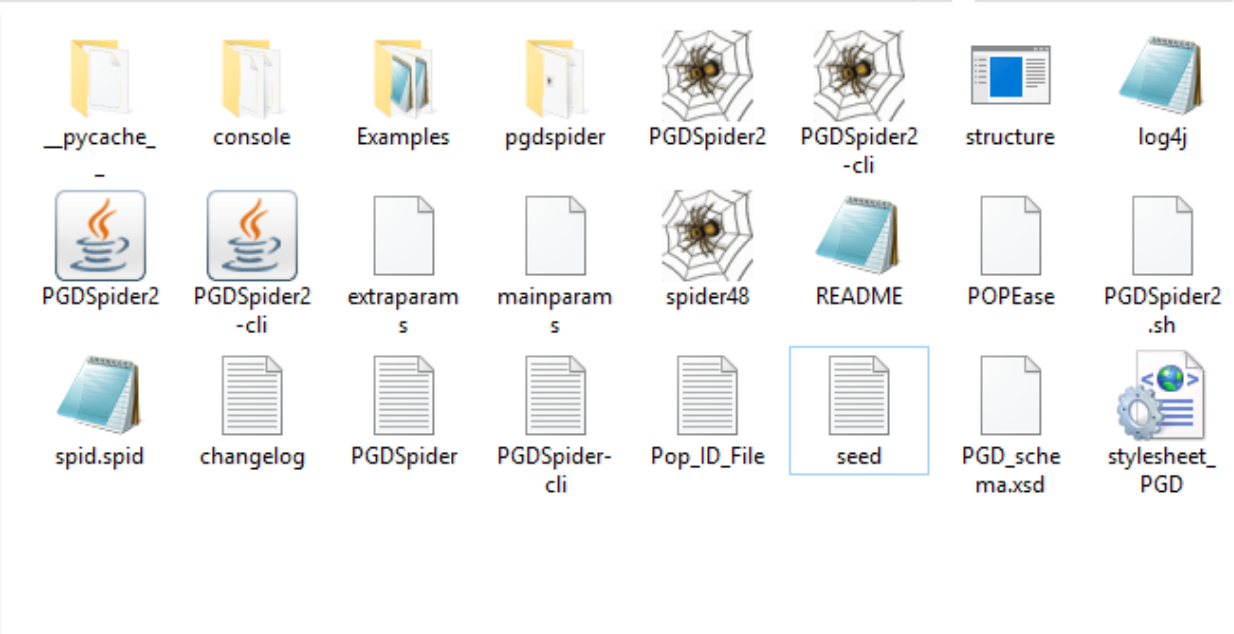
1) It is highly recommended that you download and use Anaconda as your command line for this program, especially if you are not familiar with working on a command line. The rest of the directions will be geared towards Anaconda users. Installing Anaconda will automatically install Python 3.0 for you if you do not have it on your system already. Download Anaconda here: https://www.continuum.io/downloads
2) Create a specific folder on your computer that will be used to deposit the required scripts and associated programs to run POPEase.
3) Download all files and scripts from the POPEase repository on GitHub and save them into the folder you created in step 1. THIS IS NECESSARY TO RUN THE PROGRAM CORRECTLY.
    a) Navigate to https://github.com/reige012/POPEase
    b) Click the "Download Zip" button on the right of the page
    c) Save these files into the folder you created in step 2. Make sure you use the POPEase.py file appropriate for your platform (Mac or Windows).

4) Download the following programs. Make sure to place the entire folder for each program into the new folder you created in step 2.
   a) PGDSpider- a file converting Software http://www.cmpg.unibe.ch/software/PGDSpider/
   b) STRUCTURE- a program to determine population structure from your data http://pritchardlab.stanford.edu/structure_software/release_versions/v2.3.4/html/structure.html
   c) JAVA: Mac Users: You may need to update or install Java on your system. http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
5) Move individual files from each of the step 4 program folders into the main folder you created in step 2. The POPEase process is much smoother if there is one single folder that contains all of the programs and their associated files.
   a) From PGDSpider: move all files, but not subfolders such as Example folder
   b) From Console folder (Structure's program folder) move three files:
      i) extraparams
      ii) mainparams
      iii) structure (unix executable file)
6) Move a copy of your own VCF file to the folder you made in step 2.
   a) Note: make sure your VCF file has a .vcf extension
7) That's the basic setup. To run the programs please follow the detailed instructions in Making POPEase Work.

MAC User created directory should contain the following files now:



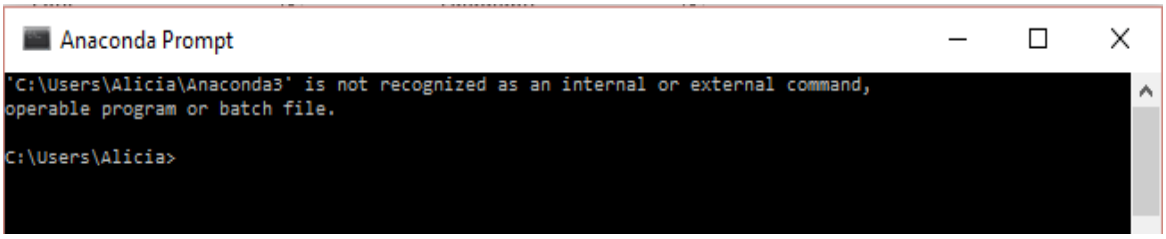
Windows User created directory should contain the following files now:



Making POPEase Work

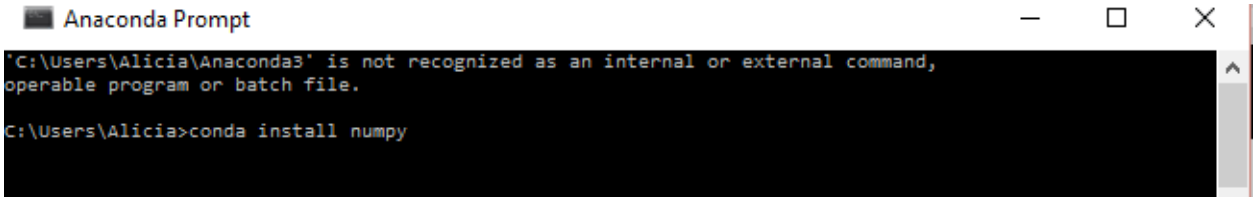
The following are the detailed instructions for running POPEase with the basic parameters. Should you like to change the parameters mentioned in Program Caveats please see Changing Parameters.

1. Windows Users: Open an Anaconda command prompt; Mac Users: Open your Terminal window

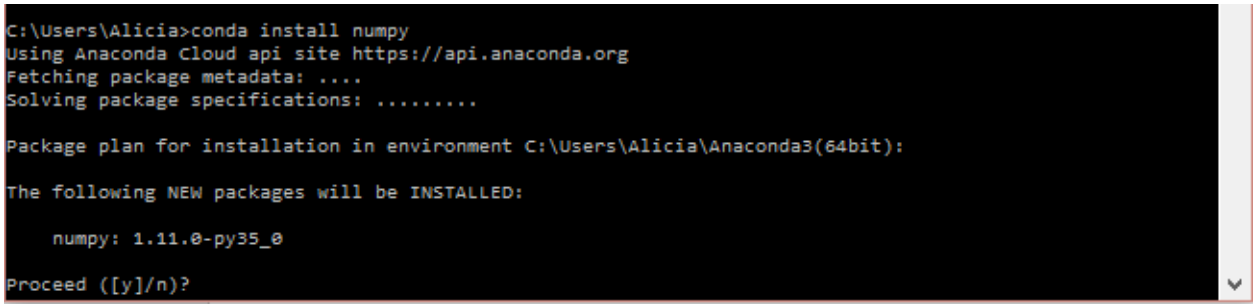


2. You will see you are located in the root directory of your computer. I'm using a windows system and this will look slightly different with a Mac system.

3. The first thing we need to do is make sure all of the dependencies are installed. To do this we use a command to install modules that are used by Python. Type the following: conda install numpy



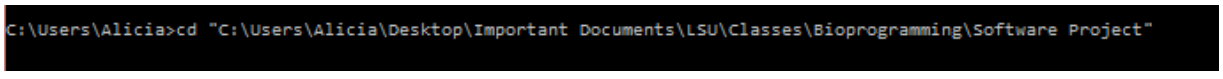
4. Press Enter. The system will search for the module NumPY. If you have the most recent version installed it will inform you, but if you need to install it you will see the system ask you for permission to import the module.



5. You should enter a "y" for 'Yes' and then press Enter again. When it is correctly installed you will see an output to the screen with an indication that it was successful.

6. Repeat steps 3-5 to install several other necessary modules using the commands:
    a. conda install pandas
    b. conda install matplotlib

7. Now let's **move** into the folder that you created for all of the program files you downloaded in Getting Started.
    a. In your command prompt type 'cd' followed by a single space then the path to the folder you created in Step 2 of Getting Started and then press Enter.
        Hint: An easy way to get the full path correct (on windows) is to click on the folder you want to use and when its highlighted drag and drop it onto your command prompt window. The path will be written automatically.



    b. Now you are located in your folder containing all of the POPEase scripts, the downloaded programs and your VCF file.

8. **Let's get started running POPEase!**

9. You'll need to make some quick alterations on two files. (<u>If you are running the example then skip step 9a., but do step 9b.</u>).
    a. Create a Pop_ID_File.txt for your data.
        i. This is simple. You need to create a basic .txt file that lists the identifier used in your VCF file for each individual, followed by a tab and then the pop_# that this individual is associated with. See below example:

File Edit Format View Help

```
Individual1     pop_1
Individual2     pop_1
Individual3     pop_2
Individual4     pop_3
```

        ii. Save the file (in the folder you made for this project) as Pop_ID_File.txt
    b. Edit the spid.spid file (or the Examplespid.spid file if doing using the Coral Example Data).
        i. This file is used by PGDSpider to identify information about your vcf data and it uses the Pop_ID_File you created above, which is why we need to make some alterations.
        ii. We simply need to alter the path to the Pop_ID_File.
            1. Open the spid.spid file
            2. Find section of the file titled: Select population definition file:VCF_Parser_Pop_File_Question=
            3. There will be a path to the Example_Pop_ID_File there. As shown highlighted in blue in the image below.
            4. Replace this path (up to the #) with the FULL path to the Pop_ID_File.txt you created in the previous step. Remember on a windows system you need to put TWO backslashes where each single backslash is located. Example: C:\Users\ → C:\\Users\\

spid.spid - Notepad

File Edit Format View Help

```
# spid-file generated: Sun Apr 24 17:26:03 CDT 2016# VCF Parser questionsPARSER_FORMAT=VCF# Do you want to
include a file with population definitions?VCF_PARSER_POP_QUESTION=true# Only input following regions
(refSeqName:start:end, multiple regions: whitespace separated):VCF_PARSER_REGION_QUESTION=# What is the
ploidy of the data?VCF_PARSER_PLOIDY_QUESTION=DIPLOID# Only output following individuals (ind1, ind2, ind4,
...):VCF_PARSER_IND_QUESTION=# Output genotypes as missing if the read depth of a position for the sample is
below:VCF_PARSER_READ_QUESTION=# Take most likely genotype if "PL" or "GL" is given in the genotype field?
VCF_PARSER_PL_QUESTION=false# Do you want to exclude loci with only missing data?
VCF_PARSER_EXC_MISSING_LOCI_QUESTION=true# Select population definition file:
VCF_PARSER_POP_FILE_QUESTION=C:\\Users\\Alicia\\Desktop\\POPEASE\\User_Created_Directory\\Examples\
\Example_Pop_ID_File.txt# Only output SNPs with a phred-scaled quality of at least:
VCF_PARSER_QUAL_QUESTION=20# Do you want to include non-polymorphic SNPs?
VCF_PARSER_MONOMORPHIC_QUESTION=false# Output genotypes as missing if the phred-scale genotype quality is
below:VCF_PARSER_GTQUAL_QUESTION=# STRUCTURE Writer questionsWRITER_FORMAT=STRUCTURE# Save more specific
fastSTRUCTURE format?STRUCTURE_WRITER_FAST_FORMAT_QUESTION=false# Specify the locus/locus combination you
want to write to the STRUCTURE file:STRUCTURE_WRITER_LOCUS_COMBINATION_QUESTION=# Specify which data type
should be included in the STRUCTURE file  (STRUCTURE can only analyze one data type per file):
STRUCTURE_WRITER_DATA_TYPE_QUESTION=SNP# Do you want to include inter-marker distances?
STRUCTURE_WRITER_LOCI_DISTANCE_QUESTION=false
```

            5. Save and close this file.

10. You'll need to decide/figure out a few important parameters before you begin.
    a. How many K's (number of populations or clusters) do you want to test?
        i. If this is a first pass over your data, I would suggest running this program for at least 2-3 K's larger than the number of populations you sampled. This will program will then test all K's from 1 to the value you selected.
        ii. POPEase requires K ≥ 2 and will return an error code if less than 2 is input.
    b. How many loci does your VCF file contain?
    c. How many individuals does your VCF file contain OR how many individuals will you be including in your structure runs?
        i. If you need to leave individuals out of the Structure run please see Changing Parameters.
    d. How many runs of Structure would you like to complete for each K value?
        i. If this is a first pass over your data a reasonable value is 10 runs. To increase accuracy for later runs you can increase this value. The more runs and K values the longer the program will take to complete, but the more accurate your results will be. POPEase requires numruns ≥ 2 and will return an error code if less than 2 numruns are input.

11. Running POPEase is simple. Call python (the program reader) followed by the name of the program file (POPEase.py for Windows; POPEase_MAC.py for Mac) followed by a series of flags that inform the program of important information. Between the flag and the input that follows it you will always need to include a space. The flags you need for this program are detailed below followed by an example on the Anaconda command prompt.

    $ python POPEase.py --runstructure--computebestk --dirpath <path to directorycontaining POPEase.py> --vcfpath <path to vcf file> --spidpath <path to spid file> --outputfile <basename to identify outputs from run> --kpop <integer value of number of clusters (K)> --loci <integer value of loci in VCF file> --individuals <integer value for number of individuals in VCF file> --numruns <integer value for number of structure runs to complete for each K value>

    a. Optional Flags:
        i. --runstructure — Structure takes a significant amount of time to run so it is possible to complete other pieces of the program without running structure. This flag MUST be present for Structure to run.
        ii. --computebestk — This flag tells the program to compute Mean_LnP(D) and Delta K values and outputs the related figures and files.
    b. Required Flags:
        i. --dirpath — Path to the directory created in Step 2 of Getting Started. This directory should contain PGDSpider and Structure programs as well as the

		POPEase.py script, Pop_ID_File.txt and your vcf file.
ii.	--spidpath	Path to the spid.spid file you altered in Step 9.
iii.	--vcfpath	Path to your VCF file
iv.	--outputfile	Base name for your output files. Can be any name you choose
v.	--kpops	Integer value of clusters (K) for structure. If unsure of this value see Step 10.
vi.	--loci	Integer value of loci in your VCF file
vii.	--individuals	Integer value of individuals in your VCF file
viii.	--numruns	Integer value of number of repetitions of Structure to complete for each value of K. Default = 10

c. Command Prompt example for the Coral Example Data Set downloaded with POPEase.

```
C:\Users\Alicia\Desktop\Important Documents\LSU\Classes\Bioprogramming\Software Project>python POPEase.py --ru
nstructure --computebestk --dirpath "C:\Users\Alicia\Desktop\Important Documents\LSU\Classes\Bioprogramming\So
ftware Project" --spidpath "C:\Users\Alicia\Desktop\Important Documents\LSU\Classes\Bioprogramming\Software Pr
oject\Examples\ExampleSpid.spid" --vcfpath "C:\Users\Alicia\Desktop\Important Documents\LSU\Classes\Bioprogram
ming\Software Project\Examples\Examplevcf.vcf" --outputfile CoralExample --kpops 7 --loci 5 --individuals 190
--numruns 10
```

d. Simply press enter after typing in all the required flags and any optional flags you desire.

12. First, the program will parse the user VCF file into a Structure-formatted file. When this is complete the program will indicate so by printing a prompt to the screen that requests the user to review the output files (for more info on output files see Program Outputs). If there is a large amount of missing data for any individuals (missing data = -9 in the Structure-formatted output) you may want to remove these individuals from the Structure run (see Changing Parameters) or remove the individuals from the input VCF file. If you would like to alter any files before running Structure you will type "N" on the command line and hit enter. This will exit the program. You can then alter your file and re-run POPEase.

    If the file appears correct type "Y" to continue running the program.

13. The screen will indicate that Structure is running. This will take a significant amount of time depending on the size of your data set, the number of clusters chosen and the number of iterations. Please DO NOT CLOSE your Anaconda prompt window because this will cause POPEase to stop running. If the curser on the screen is blinking the program is running. Be patient.

14. Upon completion of Structure the program will compute Mean_LnP(D) and Delta K if you added the --computebestk flag; if you did not, the screen will indicate that structure is complete and will exit the program.

15. After Mean_LnP(D) and Delta K have been computed the program will be finished and will exit.

16. For a list of successful outputs from POPEase see Program Outputs. If you desire a graphical display of your Structure cluster data we suggest you use CLUMPAK as their process is simple and the structure output files from POPEase are correctly formatted to work with CLUMPAK. Simply create a zip file of all of your Structure_run_f files and follow instructions on the CLUMPPAK website.

Program Outputs

All outputs from this program will be located in a Structure_Directory (made by POPEase) and located in the directory you created in Getting Started Step 2.

Outputs from PGDSpider

1. Three Files:
   a. PGDStructstderr.txt: Standard error file. Will be empty if the program worked correctly, and will contain error codes if there were any.
   b. PGDStructstdout.txt: Standard output file will contain all output from running PGDSpider. Will be empty if the program did not work correctly.
   c. STRUCTUREformatfile.txt: This is the Structure-formatted file that will be used as input for Structure.

Outputs from Structure

1. Number of files is dependent on how many clusters (--kpops) you chose to run and how many iterations (--numruns) of Structure were run on each cluster. However, there should be 1 of the following for each cluster and iteration.
   a. STRUCTUREstdout_K-value_Iteration#
   b. STRUCTUREsterr_K-value_Iteration#
   c. --outputfile_STRUCTURERUNKvalue_Iteration#_f
      i. This file begins with the --outputfile base name you choose when running the program in the anaconda prompt and always ends in _f. The _f is important for subsequent steps in the program so do not alter the file names.

Outputs from Compute Best K

1. Four files:
   a. Best_K_Analysis.csv: A CSV file of calculations completed while computing the MeanLnP(D) and Delta K values
   b. Best_K_Figures.pdf: A PDF file containing figures for MeanLnP(D) and Delta K
   c. DeltaK_figure.png: An image file of the Delta K graph
   d. Mean_LnP(D)_Figure.png: An image file of the Mean_LnP(D) graph.

e. Deter_Best_K_info.txt: A file containing all output related to choosing the best K value. The information in this file is based solely on computation and user should consult Structure documentation as well as outside resources and carefully examine their dataset before choosing a final K value to represent their data.

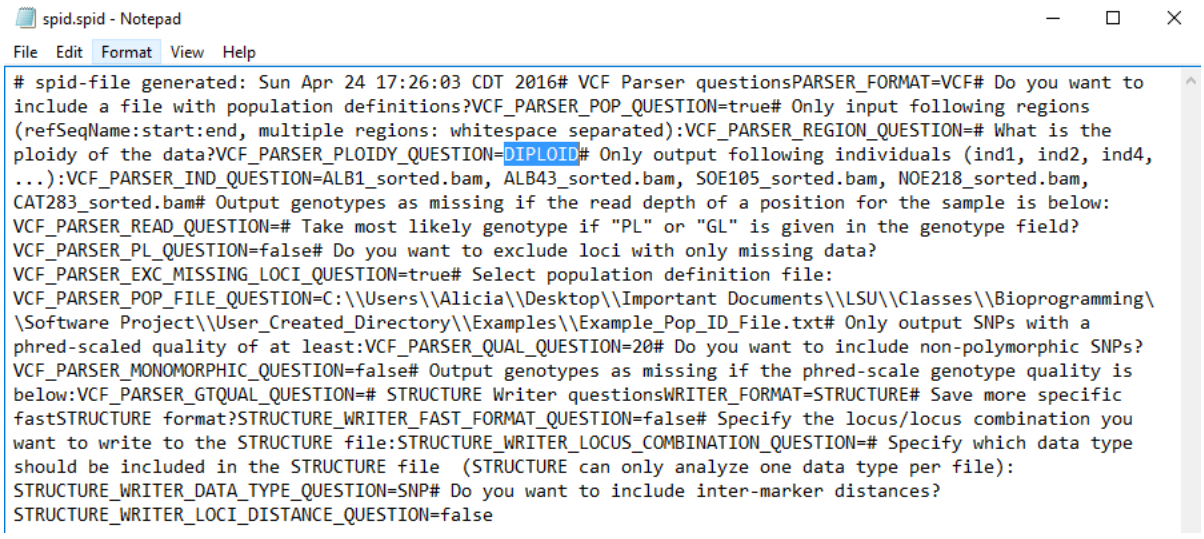
Changing Parameters

Changing some of the key parameters in the program can be a bit tricky, but if you're willing to dig a bit into the spid.spid file it shouldn't be too difficult. You've already changed the path to the Pop_ID_File.txt in your spid.spid file (Step 9 in Making POPEase Work) and this is just more of the same.

Changing from Diploid to Haploid

If you recall POPEase is defaulted to run for diploid data only, but its fairly simple to switch this to haploid.

1. Open the spid.spid file.
2. Look for: *What is the ploidy of the data?VCF_PARSER_PLOIDY_QUESTION=DIPLOID*

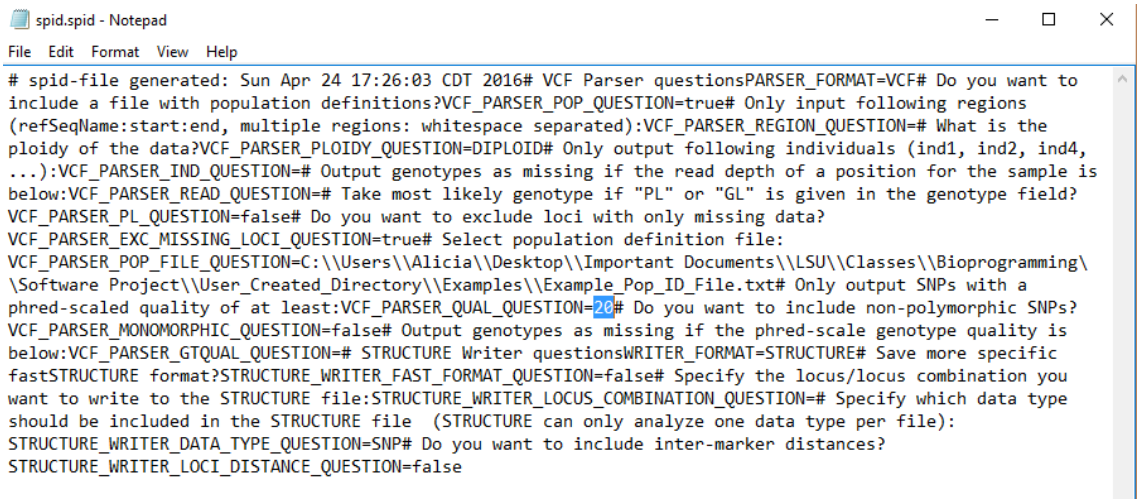


3. Replace "DIPLOID" with "HAPLOID"
4. Save the spid.spid file.
5. Run POPEase as normal.

Changing Phred Quality Score

The Phred Quality score is defaulted to exclude SNPs that have a quality score <20 (99% accuracy). Follow these instructions to make this more or less stringent.

1. Open the spid.spid file.
2. Look for: *Only output SNPs with a phred-scaled quality of at least:VCF_PARSER_QUAL_QUESTION=20*



3. Replace “20” with the desired integer. (Phred Score Table)

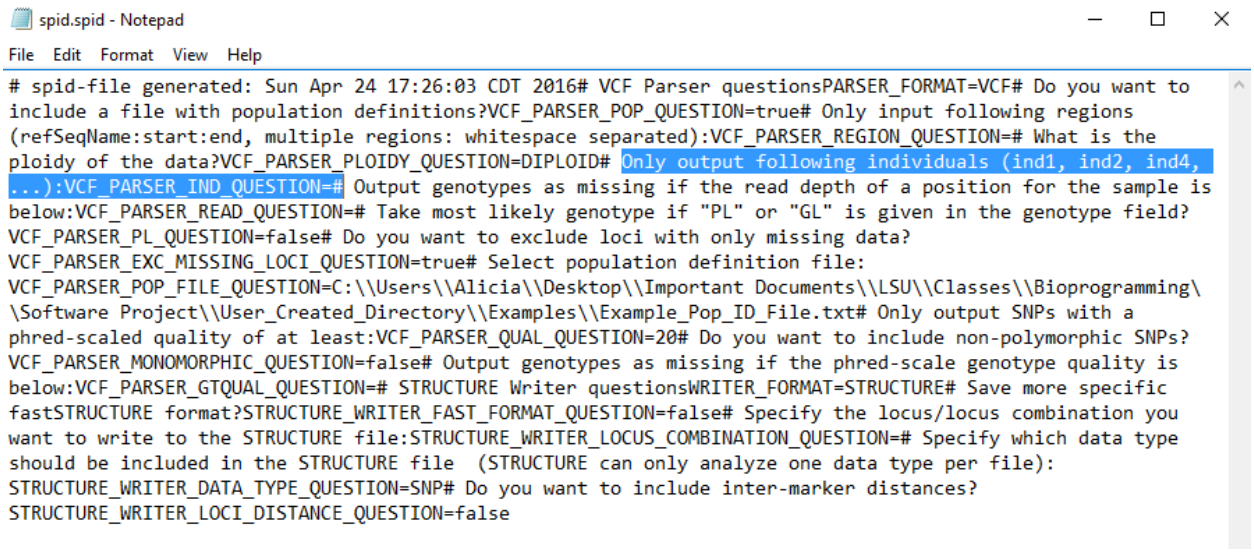
Phred Quality Score	Probability of incorrect base call	Base call accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10,000	99.99%
50	1 in 100,000	99.999%
60	1 in 1,000,000	99.9999%

4. Save the spid.spid file.
5. Run POPEase as normal.

Only Include Some Individuals in Structure Runs

If you would like to pick and choose a subset of individuals to run through Structure this is possible by editing the spid.spid file without altering your actual VCF file. You may want to do this for a number of reasons: some individuals have a lot of missing data, some individuals should not have been included initially, some individuals may skew results, you only want to examine a smaller group, etc.

1. Open the spid.spid file.
2. Look for: *Only output following individuals (ind1, ind2, ind4, ...): VCF_PARSER_IND_QUESTION=#*
3. Between the = and the #, add the list of individuals to run through structure by their name (must match their VCF file and Pop_ID_File.txt designation). Use commas to separate individuals.



4. Save the spid.spid file.
5. Run POPEase as normal.

Contact Information/Questions

For questions, concerns or information related to POPEase please contact:
Alicia Reigel: areige1@lsu.edu
Ph.D. Candidate
Louisiana State University