CRUMBLER Version 1.0 08/28/2018

Crum, T.E.[1], Schnabel, R.D.[1,2], Decker, J.E.[1,2], and Taylor, J.F.[1]. (2018). CRUMBLER: A tool for the Prediction of Ancestry in Cattle. https://doi.org/10.1101/396341

[1]Division of Animal Sciences, University of Missouri, Columbia, MO, USA 65211
[2]Informatics Institute, University of Missouri, Columbia, MO, USA 65211

Background

This pipeline was developed to integrate publically available software PLINK, EIGENSOFT, and SNPweights using python based programming to determine ancestry of a set of genotypes based on a predetermined set of reference populations.

Questions

Email: Tamar Crum, tamar.crum@mail.missouri.edu

Software

Plink v1.9 must be installed for use.
https://www.cog-genomics.org/plink2
(C) 2005-2016 Shaun Purcell, Christopher Chang GNU General Public License v3

SNPweights version 2.1 must be installed for use.
Download: https://cdn1.sph.harvard.edu/wp-content/uploads/sites/181/2014/05/SNPweights2.1.tar.gz

EIGENSOFT must be installed for use.
Git Page: https://github.com/DReichLab/EIG

EIGENSOFT packages CONVERTF and SMARTPCA is required for use by SNPweights. However, versions of SMARTPCA included in EIGENSOFT 5.0.2 and beyond are not compatible with SNPweights. The following edits must be made to the SMARTPCA source code:

1. Download EIGENSOFT from https://github.com/DReichLab/EIG/
2. Go to directory /src/eigensrc/
3. Open smartpca.c
4. Find the string: printf("trace: %9.3f\n", y) ;
5. Remove the comment characters to make the code an active line

In version 7.2.1, the code is line #1079. In version 6.1.4, the code to uncomment is line #1138. The SMARTPCA program still calculates trace, changing the code outputs the value of trace as a line in the output for use by SNPweights software.
To recompile the modified code, follow instructions by EIGENSOFT authors. This is found in the README in EIGENSOFT download base directory (https://github.com/DReichLab/EIG/blob/master/README).

Pipeline Parameter File

pipeline_par_file.par

The main parameter file for pipeline initiation. An example file is found in the example directory (pipeline_par_file_example.par). All parameters are explained below:

genotypes:
marker_list:
plink_species_ID:
assay_plink_map_files:
output_table_name:
reference_set:
number_of_chrom:
number_ref_populations:
snpweights_filename:
eig_par_file_directory_path:
SNPweights_software_path:
EIGENSOFT_software_path:

Parameters Explained

Parameter	Description	Example
genotypes:	The full path to the directory where PLINK PED genotypes are located. Only include the genotype files to be analyzed for the pipeline iteration in this directory (other files will cause errors in the program)	/example_run/example_reference_genos/
marker_list:	Path and filename of the list of markers to use in the analysis. If this is a reference analysis, a SNP list must be specified. For unknown, analyses the markers can be extracted from the already calculated SNP weights file. This uses PLINK's extract marker feature based on either marker name or marker ranges. User will specify on prompt which option they prefer based on the list provided. When asked, S = filter on SNP IDs, R = filter on marker position ranges.	/example_run/marker_list_example.csv
output_table_name:	The name of the pipeline output table (see output for more information). This parameter is not needed if reference_set is Y.	example_out.txt

plink_species_ID:	PLINK compatible species identifier for your data. If no species ID available, specify “NA”.	cow
number_of_chrom:	The number of chromosomes (autosomes). The X is assumed to be num + 1 and Y num + 2.	29
assay_plink_map_files:	Path and filename for the MAP list file (see MAP List File for more information on format)	/example_run/map_files_example.txt
reference_set:	Y: If this is reference population samples and SNP weights need to be calculated N: If this is an analysis where ancestry needs to be predicted for the samples	Y
number_ref_populations:	If reference_set is Y: must specify the number of populations in the reference set. This is the value of K.	2
snpweights_filename:	Name of the SNP weights file. If reference_set is set to Y, then this is the name of the output. If reference_set file is set to N, then this is the path and filename of the SNP weights file for the reference population you wish to use.	(if reference_set: Y) test.reference (if reference_set: N) /example_run/example_results/test.reference
eig_par_file_directory_path:	The path and directory where the par files for the EIGENSOFT packages (see EIGENSOFT Par Files for more information)	EIG_par_files/
SNPweights_software_path:	The path for running the SNPweights software.	/usr/local/bin/SNPweights2.1/
EIGENSOFT_software_path:	The path to the base directory for running the version of EIGENSOFT with the appropriate updates to smartpca for SNPweights compatibility. **Keep the EIGENSOFT directories as they appear in the original software**.	/data/usr/edited_eig/

Input Files

Genotypes Format

All genotype files must be in PLINK PED format and include appropriate accompanying files (MAP file, see plink documentation for formatting). If using multiple genotyping assays, include only one assay per genotype input file.

Note: If reference analysis is to be ran, the 6th column of the PED file should contain the population of the individual. The SNPweights software will check for this population in the 6th column to determine the number of PCs and % ancestry groups needed based on this column. The 6th column, should never, contain the values “-9”, “9”, or “0”. If no population or phenotype is needed (for unknown analysis), the 6th column can be set to “1”.

Genotypes should be coded in AB or 12 format.

Genotype and MAP File Nomenclature

The input files must be named following these rules:

PED

[your chosen identifier].[assay].ped

MAP

[your chosen identifier].map

Example Reference Genotype Files: /example_run/example_reference_genos/

Example Unknown Genotype Files: /example_run/example_unknown_genos/

Marker List

The user must specify a list of SNPs to use in the analysis for all reference analyses. If multiple assays are used and iterations of the pipeline will be compared, using the same set of SNPs is important. The SNPs are filtered using the PLINK extract a subset of SNPs feature. If user wishes to extract SNPs based on SNP/marker name, a text file must be created where the file is just a list of SNPs, one per line.

```
snp001
snp004
snp201
```

Additionally, the user can choose to extract markers based on a range of positions using PLINK –extract [markerlist.txt] –range. The marker file must be in the format of one range per line containing a column for each of the below, whitespace separated (see plink manual for more information).

CHR

BP1 (start of range, physical position in base units)

BP2 (end of range)

Label (name)

The user will be prompted following initiation of the run command to specify whether they wish to filter on SNP IDs (S) or Ranges (R). It should be noted for optimal results, if multiple assays and multiple runs are used, filtering on SNP ID will be more consistent, as the number of markers pulled from each assay will be the same.

If no list is specified and the analysis is for unknown samples. The markers indicated in the SNP weights file specified by the user in the parameter file will be extracted and used for analysis.

For further clarification on the filtering process, see PLINK documentation.

Example Marker List: /example_run/marker_list_example.csv

MAP List File
A MAP file list must be created. The list should be saved, 1 assay per line, as a text file. If only one assay is being used, there will be only 1 line in the file. The list must take on the following format:

[assay]: path and file name of corresponding MAP including .map extension

Note: [assay] must match the PED file. This is how the program knows which MAP file corresponds with a genotype file.

Format Example:
Assay1: /home/example/map_files/example1.map
GGL4: /home/example/PLINK_files/GGL4_2017.map

Example MAP List File: /example_run/map_files_example.txt

EIGENSOFT Par Files
EIGENSOFT requires a par file for each of its packages. All the par files used for this pipeline are located in the directory /par_files/. Keep the directory together. Each par file is explained below. The inputs given are required. **Do not edit lines shown in red!** Other lines can be edited per your needs. See EIGENSOFT package readme(s) to assess any changes/additions you want to make. EIGENSOFT does have additional arguments for input, if you wish to use them add the appropriate information to the bottom of the appropriate par file. Ensure no additional blank lines are trailing the par file.

A directory containing the specific EIGENSOFT par files for the run will be generated and contained within the base directory of which CRUMBLER was run.

File: par.reference.PED.EIGENSTRAT
Convert PED format to EIGENSTRAT format using CONVERTF package for reference population genotypes.

```
genotypename:    ref_merged/ref_input.ped
snpname:         ref_merged/ref_input.map
indivname:       ref_merged/ref_input.pedind
outputformat:    EIGENSTRAT
genotypeoutname: ref_eigenstrat/ref.eigenstratgeno
snpoutname:      ref_eigenstrat/ref.snp
indivoutname:    ref_eigenstrat/ref.ind
familynames:     NO
outputgroup:     YES
numchrom:        [auto]
```

File: par.reference.smartpca

Run SMARTPCA for reference population samples.

```
genotypename:    ref_eigenstrat/ref.eigenstratgeno
snpname:         ref_eigenstrat/ref.snp
indivname:       ref_eigenstrat/ref.ind
evecoutname:     ref_eigenstrat/ref.evec
evaloutname:     ref_eigenstrat/ref.eval
altnormstyle:    YES
numoutevec:      [auto]
numoutlieriter:  0
numchrom:        [auto]
```

File: par.reference.calc_snpwt

Calculate SNP weights based on reference population samples and the SMARTPCA output.

```
geno: ref_eigenstrat/ref.eigenstratgeno
snp:  ref_eigenstrat/ref.snp
ind:  ref_eigenstrat/ref.ind
evec: ref_eigenstrat/ref.evec
eval: ref_eigenstrat/ref.eval
log:  ref_eigenstrat/ref.log
snpwtoutput: [auto]
```

File: par.unknown.PED.EIGENSTRAT

Convert PED format to EIGENSTRAT format using CONVERTF package for unknown sample genotypes.

```
genotypename:    unk_merged/unk_input.ped
snpname:         unk_merged/unk_input.map
```

```
indivname:       unk_merged/unk_input.pedind
outputformat:    EIGENSTRAT
genotypeoutname: unk_eigenstrat/unk.eigenstratgeno
snpoutname:      unk_eigenstrat/unk.snp
indivoutname:    unk_eigenstrat/unk.ind
familynames:     NO
outputgroup:     YES
numchrom:        [auto]
```

File: par.inferancestry

Predict ancestry for unknown samples based on the SNP weights file calculated for the reference population samples.

```
geno: ./unk_eigenstrat/unk.eigenstratgeno
snp:  ./unk_eigenstrat/unk.snp
ind:  ./unk_eigenstrat/unk.ind
snpwt: [auto]
predpcoutput: [auto]
```

Running Pipeline

Keep the configuration (run_CRUMBLER.py) file and all the source code in the same directory and run the program from it.

```
python [insert path if one is needed]/src/run_CRUMBLER.py pipeline par file
```

Output File

There are two different outputs created containing the breed compositions of the desired samples. See SNPweights manual for information that is contained in the original raw output of the software. This pipeline is designed to manipulate the raw output to text file with a format that is easily read and stored.

sample IDs	# of markers used for inference	% assigned to breed 1	...	% assigned to breed n	name and path of marker list used	name and path of reference SNP weights file used

The rows are the samples that breed composition was desired (row number = n unknown samples + 1). The first row contains a header that specifies the content of each column. The % assignment columns will be equal to the number of reference populations in the SNPweights file used for calculation. The population names that were included in the creation of the reference SNP weights file are used as the labels for the % ancestry columns.

Example Output: /example_run/example_results/example_out.txt

Testing Software

Example files are indicated in /example_run/ folder in the pipeline directory. To run, the test keep all contents of folder together and run within the directory (copy entire directory to a working environment and run within the directory in that environment). To ensure your software is running correctly, check your results with the results located in example_run/example_results/.

To run test, update the file paths (eig_par_file_directory_path, SNPweights_software_path, EIGENSOFT_software_path) in the par files (example_reference_par.txt and example_unknown_par.txt).

To run reference test (within example_run direcotry):

```
python {insert path}/src/run_CRUMBLER.py example_reference_par.txt
```

Output Files/Directories:
ref_eigenstrat/
ref_filtered/
ref_merged/
test.reference - This should match the file in example_results/.

To run unknown test (within example_run):

```
python {insert path}/src/run_CRUMBLER.py example_unknown_par.txt
```

Output Files/Directories:
unk_eigenstrat/
unk_filtered/
unk_merged/
example_out.txt - This is where you results are! Should match the results in example_results/.

Copyright/Licenses/Citations for Integrated Software

CRUMBLER:
Released under GNU GPL. See Github page for information on CRUMBLER and Third Party Licenses https://github.com/tamarcrum/CRUMBLER.

EIGENSOFT:
The EIGENSOFT package implements methods from the following 3 papers:
Patterson et al. 2006 PLoS Genet (population structure)
Price et al. 2006 Nat Genet (EIGENSTRAT stratification correction)

Galinsky et al. 2016 Am J Hum Genet (FastPCA and PC-based selection statistic)

SNPweights:
Chen et al. "Improved ancestry inference using weights from external reference panels", Bioinformatics (2013), 29, 1399-406.

PLINK:
Package : PLINK v1.90b3.31
Authors : Shaun Purcell, Christopher Chang
URL   : www.cog-genomics.org/plink/1.9/

Chang CC, Chow CC, Tellier LCAM, Vattikuti S, Purcell SM, Lee JJ (2015) Second-generation PLINK: rising to the challenge of larger and richer datasets. GigaScience, 4.