CallSpecies.py : Automated Species Calling for GT-seq Pipelines

Zachary L. Robinson
zrobinson@critfc.org

I. Overview

Purpose: Implement a simple scoring system to make automated species calls using species-informative loci that possess opposite homozygotes among species. The implementation was intended to incorporate seamlessly into commonly used GT-seq bioinformatic pipelines and provide species calls along with genotypic data.

Approach: This script was written to accompany the GT-seq genotyping pipeline described in Campbell et al. (2015) and add species calls to the standard output from *GenoCompile.pl*. The user specifies species-informative loci, the species association with each allele, and provides a scoring weight to each locus in the species-seq file. The script will consider all loci provided and make a consensus species call based on the species that achieves the highest score for each sample (discussed in more detail below). Key design features of this script include: 1) operates on a variety of species-specific GT-seq panels because it does not require that genotypic inputs possess a consistent set of loci (species-informative or otherwise) 2) the species scoring system accommodates loci that are not diagnostic but include/exclude a subset of species, 3) it is robust to missing data and low-frequency shared polymorphisms.

II. System Requirements

Python Version: The distributed Python scripts were written and tested in Python v3.8.12 through v3.6.5 on MacOS and Linux systems. On Unix-based operating systems, please confirm an appropriate version of Python is installed and referenced in the shebang (#!) line of the scripts. For Windows users, please consult Python documentation for trouble-shooting advice. (https://docs.python.org/3/using/windows.html#shebang-lines)

Python Modules: The CallSpecies.py script only requires one python module `argparse`, which is part of the standard module library. The two additional scripts for simulating missing data and shared polymorphisms require the python modules `argparse`, `random`, and `scipy`. The scripts should operate normally with the prerequisites satisfied and executable permissions granted.

III. Quick Start

After satisfying system requirements and dependencies, lets confirm things are working appropriately with simulated data.

```
>./MakeMissingGenos.py --help
>
>./MakeMissingGenos.py --SpeciesSeq SpeciesSeq.csv --outGENOS MissingTEST-
Genos.csv --iters 5
>./CallSpecies.py --help
>
>./CallSpecies.py --SpeciesSeq SpeciesSeq.csv --inGENO MissingTEST-Genos.csv --
outGENO MissingTEST-Genos_CALLED.csv
```

IV. CallSpecies.py: Named Arguments

`--SpeciesSeq` `SpeciesSeq file path:`
`Locus,A1,A2,SpeciesA1,SpeciesA2,Weight`

Description:
The species-seq file provides the genotypic associations with each species and locus-specific scoring weight. The file is comma-separated and is assumed to have a header line, which is ignored. Each row in the file corresponds to a species-informative locus. The first column is the locus name and must match the genotypic data exactly. The second and third columns are the two nucleotides (alleles) associated with the biallelic SNP. The fourth and fifth columns are the semi-colon delimited string of species associated with allele 1 and allele 2, respectively. The sixth column is the locus-specific scoring weight that will be added or subtracted from each species score in column 4 and 5. For an example, see the *SpeciesSeq.csv* file distributed with this script. Species and loci can be added and removed at will from this file, however, it is advised that you use the provided simulation scripts to access the effects of your changes.

`--inGENO` `File path for wide-format, comma-separated genotypic input file i.e., GenoCompile.py derived file.`

Description:
The input genotype file path for which species calls are desired should be comma-separated and in wide-format (i.e., one row per individual). The expected genotype format is a colon-delimited string (e.g., A:A), and missing data coded as 0:0. The script is naïve to metadata columns and uses the header line of the genotype file to determine the column positions of species-informative markers. Therefore, the only requirement is CSV-format, loci names in the header line match the species-seq file, and genotypes are colon-delimited alleles.

`--outGENO` `File path for genotypic output file with species calls fields added.`

Description:
The output file path with species information appended. The only modification to input is the addition of two columns. The SpeciesCall column which is ideally a single species name. The second column is SpeciesHET, which reports the number of heterozygous genotypes encountered and number of successfully genotyped species-informative markers as a semi-colon delimited string (i.e., 1;54).

--thresHET Threshold level of individual heterozygosity at species markers after which the species call is flagged with 'ReviewHET;' default: 0.06

Description:
As previously stated, the scoring system is designed for opposite homozygotes among species. If heterozygosity is above this threshold for species-informative markers, the species calls is appended with 'ReviewHET;' (e.g., ReviewHET;Omy).

--thresMS Threshold proportion of maximum absolute score that at least one species must obtain to make a species call for a sample. default: 0.5

Description: The absolute maximum score that a species can achieve is defined by the species-seq file and assumes no missing data or heterozygous genotypes. This the default value of 0.5 requires that one species acquires a species association score that is at least 50% of a "perfect" score for that species.

--buffMP The buffer range around the highest proportion of maximum possible score (given missing data and heterozygotes) to consider a species as a candidate for a sample. default: 0.001

Description:
This is simply to allow approximate ties among species. If a tie occurs the species call will be a semi-colon delimited string (e.g., Omy;Ocl).

--pruneMS Threshold for proportion of maximum absolute score to retain a species as a candidate for a sample. default: 0.34

This option was intended to deal with species that are supported by few loci in the species-seq file. For example, a species that did not pass --thresMS may be reported along with the highest scoring species because it achieved a high proportion maximum possible score. For example, a species that is described by three loci in the species-seq file and genotypes successfully at one locus may be retained as a candidate because it achieves a maximum possible score based on that single locus (i.e., 1/1), but would be removed using the default pruning value based on absolute maximum score (i.e., 1/3).

--colSTRT 1-based column position to insert two species call fields in wide-format genotypic input. default: 5

Description:
Where do you want the SpeciesCall and SpeciesHET columns added to the input genotype file?

--outScoreMat If specified, output file path for species score matrix (Optional).

Description:
Returns the information used to make species determinations. The first row is the absolute maximum score for each species as defined by the species-seq file. The next rows are the scores that would be achieved for representative genotypes for each species as provided by the species-seq file. The following rows are the samples provided by --inGENO. Each score after the first row, is a semicolon-delimited string that provides the score obtained by the species and the maximum possible score that could be obtained given missing data or heterozygous genotypes.

--outRepGeno If specified, output file path for representative genotype per species (Optional).

Description:
Provides the expected genotype for each species as specified in the species-seq file in wide-format. May be useful for reviewing species-informative markers.

V. *CallSpecies.py: Scoring Method Visualized*

Step 1: Unknown Species Multi-locus Genotype is Provided

Species-informative Homozygous Loci	Loc-1	Loc-2	Loc-3	…
Unknown Species Genotype	C:C	G:G	A:A	…

Step 2: Species Association Scores are Calculated

Species associations and weights are extracted from SpeciesSeq.csv.

Locus	A1	A2	SpeciesA1	SpeciesA2	Weight
Loc-1	A	C	Sp2;Sp3;Sp4	Sp1	2
Loc-2	G	C	Sp1;Sp4	Sp2;Sp3	1
Loc-3	A	T	Sp1;Sp3	Sp4	2
…	…	…	…	…	…

Scoring weights are added or subtracted for each candidate species (Sp1-Sp4) to obtain species association scores for the unknown species genotype.

	Sp1	Sp2	Sp3	Sp4
Loc-1	+2	-2	-2	-2
Loc-2	+1	-1	-1	+1
Loc-3	+2	0	+2	-2
…	…	…	…	…
Species Association Score	76	3	50	30

Step 3: Species Call is Made

Definitions:

Absolute Maximum Score: maximum achievable score for each species based on all species-informative loci genotyping successfully (i.e., the "perfect score").

Maximum Possible Score: Maximum possible score for each species given that some loci may not have genotyped successfully or are heterozygous in a particular sample.

Satisfy *--thresMS*

At least one species must pass *--thresMS* (default=0.5) or SpeciesCall='NoCall'

Sp1 and Sp3 pass the threshold.

	Sp1	Sp2	Sp3	Sp4
Absolute Maximum Score	77	9	73	85
Species Association Score	76	3	50	30
Proportion Max Absolute Score	0.99	0.33	.68	.35

Satisfy *--buffMP*

The highest proportion of max possible score is 1.00 (Sp1 and Sp2). Sp3 and Sp4 are not within --buffMP (default:10^{-3}) of the top score and are dropped as candidates.

	Sp1	Sp2	Sp3	Sp4
Maximum Possible Score	76	3	72	85
Species Association Score	76	3	50	30
Proportion Max Possible Score	1.00	1.00	.69	.35

Satisfy *--pruneMS*

Sp2 does not pass the *--pruneMS* threshold (default: 0.34) for Proportion Max Absolute Score and is dropped as a candidate.

SpeciesCall='Sp1'

VI. Simulate Missing Genotypes Based on *SpeciesSeq.csv* File

Description:

MakeMissingGenos.py determines the number of loci with a specified genotype for each species (i.e., are informative). It then simulates all levels of missing data {$n_{loci,}$..,1} for a specified number of iterations. The purpose of this script is to determine if a systematic error may occur with the user-defined species-seq file and selected settings in *CallSpecies.py*. As a word of caution, the number of iterations is for each level of missing data. The resulting file has approximately $N_{species}$ * N_{loci} * *--iters* genotypes.

Example Run:

```
>./MakeMissingGenos.py --help
>
>./MakeMissingGenos.py --SpeciesSeq SpeciesSeq.csv --outGENOS MissingTEST-
Genos.csv --iters 5
```

Named Arguments:

--SpeciesSeq SpeciesSeq file path:
Locus,A1,A1,SpeciesA1,SpeciesA2,Weight

--outGENOS File path for wide-format, comma-separated genotypic output file i.e., GenoCompile.pl formatted file

--iters The number of random samples at each level of missing genotypic data for each species. Values greater than 1000 are likely unnecessary and create a large file.

VII. Simulate Shared-Polymorphisms Based on *SpeciesSeq.csv* File

Description:
MakeSharedPoly.py will use the SpeciesSeq.csv file and simulate genotypes for each species with an alternate allele frequency (i.e., the allele not associated with that species). The purpose of the script is to investigate the sensitivity of the SpeciesSeq.csv file configuration and the settings used in *Callspecies.py* to shared-polymorphisms among species and minor contamination. The number of genotypes (rows) produced is approximately $N_{iters} * N_{EffSpecies} * N_{nLociPoly} * N_{altFREQ}$.

Example Run:

```
>./MakeSharedPoly.py --help
>
>./MakeSharedPoly.py --SpeciesSeq SpeciesSeq.csv --outGENOS
SharedTEST-Genos.csv --iters 5
```

Named Arguments:

--SpeciesSeq SpeciesSeq file path:
Locus,A1,A1,SpeciesA1,SpeciesA2,Weight

--outGENOS File path for wide-format, comma-separated genotypic output file i.e., GenoCompile_v5.1.py formatted file

--iters The number of simulated individuals for each species, nLociPoly, and altFREQ. Values much greater than 1000 are likely unnecessary and create a large file default: 100

--EffSpecies A single species to be affected by the shared polymorphism or 'all' to affect all species default: all

--nLociPoly The number of loci with a shared polymorphism among species. Integer or comma-separated string of integers default: 1,2,5

--altFREQ The frequency of the allele not associated with the species. Float value or comma-separated string of float values default: 0.01,0.05,0.1