TESS3 reference manual
Command-line program and R wrapper functions

Kevin Caye (kevin.caye@imag.fr)
Olivier François (olivier.francois@imag.fr)

June 18, 2015

Please, print this reference manual only if it is necessary.

Summary

Geography is an important determinant of genetic variation in natural populations, and its effects are commonly investigated by analyzing population genetic structure using spatial ancestry estimation programs. Classical spatial ancestry estimation programs do not scale with the dimension of the data sets generated from modern sequencing technologies, and more efficient algorithms are needed to analyze genome-wide patterns of population genetic variation in their geographic context.

The computer program TESS3, which has functionalities similar to previous versions of TESS [?], has run-times several order faster than those of common Bayesian clustering programs. In addition, the program can be used to perform genome scans for selection based on ancestral allele frequency differentiation, and to separate non-adaptive and adaptive genetic variation.

This documentation aims to help users to run the TESS3 command-line engine on Linux, Mac and Windows operating systems, and presents some R functions that facilitate the post-processing of the program outputs. The main features of TESS3 are illustrated using an example data set, simulated from European lines of the plant species *Arabidopsis thaliana*.

1 Program installation

The installation of TESS3 requires that CMake, a the cross-platform open-source build system, is installed on your computer OS (see http://www.cmake.org/). The TESS3 program source code can be downloaded from the following Github address https://github.com/cayek/TESS3. After downloading files as a zipped archive, unzipping the archive will create a TESS3-master directory on your system. The directory contains

- src : the source code of the program,
- external: external libraries,
- examples: R scripts that run example data analysis with the TESS3 program.
- doc: program documentation.
- data : some simulated data sets for experimenting the program use.

The installation of TESS3 can be performed by typing the following instructions from a terminal session

```
mkdir build/
cmake -DCMAKE_BUILD_TYPE=release ../
cd build/
make TESS3
```

The executable program file TESS3 can be found in the build directory.

2 Data format

2.1 Input file

TESS3 requires two input files, a first one recording individual genotype data and the second one containing the geographic coordinates of each sampled individual. For organism genomes of arbitrary ploidy, the standard data type for TESS3 is the **single nucleotide polymorphism** (SNP) type. The genotype matrix must be formatted in the **geno** format and the coordinate file must be formatted in the **coord** format.

Users that want to process allelic data, such as microsatellite markers or AFLPs, and have their data in the TESS 2.3 format can also use TESS3. They need to convert their data in the geno+coord data format, and can do this using the `tess2tess3` function provided with the R scripts in the `example` directory.

- **geno** (example.geno)
  The **geno** format has one row for each SNP. Each row contains 1 character per individual. For diploid genomes, 0 means zero copies of the reference allele, 1 means one copy of the reference allele, 2 means two copies of the reference allele, and 9 codes for some missing data. Here is an example of a geno file for $n = 3$ individuals and $L = 4$ loci.

```
112
010
091
121
```

- **coord** (example.coord)
  The **coord** format has one row for each individual. Each row contains the `longitude latitude` information of each individual.

```
2.5154 5.4390
-8.4293 4.0197
1.3536 5.5852
```

Users having their genotype data in the **ped**, **ancestrymap**, **vcf** or **lfmm** format can use the R package to the LEA to convert them in the **geno** fromat [**?**].

2.2 Output files

TESS3 produces three main **output files**.

- The file with the extension the **.K.Q** contains the Q-matrix of individual ancestry coefficients. The Q-matrix has n rows (the number of individuals) and K columns (the number of ancestral populations).
- The file with the extension **4K.G** contains the G-matrix of ancestral genotype frequencies. The G-matrix has $N_g \times L$ lines (the number of genotypes times the number of SNPs) and K columns (the number of ancestral populations). For diploid organisms, the first row represents the frequencies of genotype 0 in each ancestral population, the second row represents the frequencies of genotype 1, and the third line the ancestral frequencies of 2.
- The file with the extension **.Fst** contains the values of the ancestral allele frequency differentiation statistic F_{ST} computed at each locus.

There are additional **output files** that are useful to get some information about the nearest-neighbor graph, the least-squares error and the cross-entropy criterion computed by the program.

- The file with the extension **.W** contains the weight matrix for the nearest-neighbor graph.
- The file with the extension the **.sum** contains the values of the least-squares error, the cross entropy criterion computed with all the data and with a percentage of masked data when the user asked the program to compute it.

3 Run the program

3.1 Command-line

The TESS3 program can be executed from a command line. The basic format is as follows

```
./TESS3 -x genotype_file.geno  -r coordinates_file.coord -K number_of_ancestral_populations
```

Only the three options `-x -r -K` are mandatory. The ordering of the options in the command line is unimportant. Here is a description of these options :

- `-x genotype_file.geno` is the path to the genotype file (in the .geno format),
- `-r coordinates_file.coord` is the path to the coordinate file (in the .coord format),
- `-K number_of_ancestral_populations` is the number of ancestral populations.

Some additional options are available :

- `-m ploidy` 1 if haploid, 2 if diploid (default : 2).
- `-a alpha` is the value of the normalized regularization parameter (by default : 0.001). This parameter controls the geographic regularity of ancestry estimates.

- `-W edge_weight_input` is the path to a file that containing a pre-specified edge weight matrix for the nearest-neighbor graph (separator = space character).The program uses this matrix when the option is set, otherwise it uses the default values.
- `-q output_Q` is a path for the output file containing the Q-matrix of ancestry coefficients. The default name of the output file is the same name as the input file with the extension `.K.Q`.
- `-g output_G` is a path to the output file containing the ancestral genotype frequencies. The default name of the output file is the same name as the input genotype file with the extension `.K.G`.
- `-f output_FST` is a path to the output file containing the values of ancestral allele frequency differentiation statistic for each loci. The default name of the output file is the same name as the input genotype file with the extension `.K.Fst`.
- `-y output_sum` is a path to the output file containing the value of the least-squares criterion, the cross entropy criteria on all data and on masked data only. By default, the name of the output file is the same name as the input genotype file with the extension `.K.sum`.
- `-c perc` is the percentage of masked genotypes. If this option is set, the cross-entropy criterion is computed (see [?] for more details on the cross-entropy criterion). The default percentage is 5%.
- `-e tolerance` is the tolerance error in the optimization algorithm (by default : 0.0000001).
- `-i iteration_number` is the max number of iterations of the algorithm (default : 200).
- `-I nb_SNPs` starts the algorithm with a run using a subset of nb_SNPs random SNPs. This option speeds up the estimation algorithm for very large data sets.
- `-Q input_Q` is the path to an initial file for the Q matrix containing individual admixture coefficients. If both `-I` and `-Q` are set, `-Q` is chosen.
- `-s seed` is a seed to initialize the random number generator.
- `-p p` is the number of CPUs to use when the algorithm is run on a multiprocessor system. Be aware that the number of processes has to be lower or equal to the number of CPU units available on your computer (default : 1).

A summary of options can be obtained by typing the following command

```
./bin/TESS3 -h
```

3.2 Example

To run a very simple example of the use of `TESS3`, consider data simulated from the chromosome 5 of the model plant species *A. thaliana*. The genotype matrix contains 170 genotypes from European lines of the plant (26943 SNPs). Since *A. thaliana* is a selfing species, the heterozygote frequency was equal to zero, and the diploid genotypes were recoded as haploid. Copy the program executable file in the `./data/simulated/Athaliana` directory and type

```
./TESS3 -x Athaliana.geno  -r Athaliana.coord -K 3 -m 1
```

Then, open a `R` session and type

```
> par( mfrow = c(2,1) )
> Qmatrix = t( read.table("Athaliana.3.Q") )
> barplot(Qmatrix, col = 2:4)
> Fst = read.table("Athaliana.3.Fst")
> plot(Fst, main = "Manhattan Plot", cex = .4)
```

Those commands visualize the ancestry coefficient matrix computed by `TESS3` and display a Manhattan plot of F_{ST} values along the fifth chromosome of the plant species.

3.3 R wrapper functions

To facilitate the interpretation of results, the `TESS3` program can be launched from the `R` command line, using the functions in the `src/Rwrapper/TESS3.R` script. Users need to have `TESS3` installed on their OS, and source this script from an `R` session. The main `R` functions are described below

- `TESS3`

  Description : A wrapper function that calls the `TESS3` command-line program. This function creates a directory named `TESS3_workingDirectory` to record all output files

Usage

```
object = TESS3( genotype,
                coordinates,
                K,
                ploidy=1,
                seed=-1,
                rep = 1,
                maskedProportion = 0.0,
                alpha = 0.001 )
```

Arguments

- `genotype` : A path to the genotype matrix in the **geno** format or a matrix of size n individuals by L loci.
- `coordinates` : A path to the spatial coordinate matrix in the **coord** format or a matrix of size n individuals by L loci, or a matrix of size n by 2.
- `K` : number of ancestral cluster. It can be a sequence of integer values. The `TESS3` program will be run for each element of the sequence.
- `ploidy` : 1 if haploid, 2 if diploid.
- `rep` : number of runs for each K value.
- `maskedProportion` : if `maskedProportion > 0`, the cross-entropy criterion is computed for this percentage of masked genotypes.
- `alpha` : value of the normalized regularization parameter. This parameter controls the spatial regularity of the ancestry estimates.

– `Getter`

Description Functions useful to fetch results of `TESS3` R function. UN PEU LEGER DETAILLER ON NE SAIT PAS CE QUE CA RETOURNE

Usage

```
getQ( project, K, run = "best" )

getG( project, K, run = "best" )

getFst( project, K, run = "best" )

getCrossEntropy( project, FUN = mean )

getLeastSquareError( project, FUN = mean )
```

Arguments

- `project` : object returned by the `TESS3` R function.
- `K` : number of ancestral cluster.
- `run` : number of the run, or `"best"` if you want the best result with respect to the least-squared criterion.
- `FUNCTION` : function used to summarize data over all run.

– `Reader`

Description Function useful to read data from file.

Usage

```
read.coord( file )
```

Arguments

- `file` name of the file to read.

4 Tutorial

4.1 Data set

We illustrate the use of TESS3 and the R functions using simulated *Arabidopsis thaliana* data as in a previous sections [?]. We consider a data set of $n = 170$ (haploid) individuals genotyped at $L = 26943$ SNPs.

4.2 Example of visualization of ancestry coefficients using TESS3

This example describes how to run TESS3 from the R command line for several values of the number of ancestral populations, and how to plot a graph of the cross-entropy criterion as function of K. It also uses scripts available from POPS web-site (http://membres-timc.imag.fr/Olivier.Francois/pops.html) to display geographic representations of ancestry coefficient maps using raster maps [?].

```
source("TESS3-master/src/Rwrapper/TESS3.R")
library(LEA)

setwd( "TESS3-master/data/simulated/Athaliana" )
###########################################################################
# Run TESS3 on a data set simulated from Arabidopsis thaliana #
###########################################################################

#read data
spatialData = read.coord("Athaliana.coord")
n = nrow(spatialData)

project = TESS3( genotype = "Athaliana.geno",
                 coordinates = "Athaliana.coord",
                 K = 1:5,
                 ploidy = 1,
                 rep = 1,
                 maskedProportion = 0.2)


###########################################
# Choose  K using the cross-entropy criterion #
###########################################

plot( 1:5,
      getCrossEntropy( project ),
      main  = "Cross entropy",
      type="b",
      xlab = "K",
      ylab = "cross entropy" )

################################
# Display results  for K = 3 #
################################

# R script available on http://membres-timc.imag.fr/Olivier.Francois/pops.html
source("POPS_direction/R/POPSutilities.r")

asciiFile="/home/cayek/Projects/TESS3/data/simulated/Athaliana/down_etopo1.asc"
grid=createGridFromAsciiRaster(asciiFile)
# To display only altitudes above 0:
constraints=getConstraintsFromAsciiRaster(asciiFile,cell_value_min=0)

maps(matrix = getQ( project, K = 3 ),
     coord = spatialData,
     grid=grid,constraints=constraints,method="max",main="ancestry coefficient with K = 3")
```

4.3 Example of a genome scan for selection using TESS3

This section shows an example of the use of TESS3 to perform a genome scan for selection based on the computation of ancestral allele frequency differentiation statistics. Here TESS3 was run with $K = 3$ ancestral populations. The F_{ST} statistics were transformed into squared t-scores and p-value using a Fisher distribution. Inflation of the test statistic caused by population structure was corrected by recalibrating the t^2-scores to follow a uniform distribution under the null hypothesis. The false discovery rate was controlled using the Benjamini-Hochberg procedure and plot the Manhattan plot.

```
source("TESS3-master/src/Rwrapper/TESS3.R")
library(LEA)

setwd( "TESS3-master/data/simulated/Athaliana" )
###########################################################################
# Run TESS3 on data  #
###########################################################################

#read data
spatialData = read.coord("Athaliana.coord")
genotype = read.geno("Athaliana.geno")
n = nrow(spatialData)

project = TESS3( genotype = genotype,
                 spatialData = spatialData,
                 K = 3,
                 ploidy = 1,
                 rep = 5 )


#############################
# Genome scan for selection #
#############################

#### Fst with TESS3
Fst = getFst( project, K = 3 )
Fst[Fst < 0.0] = 0.0

#### Convert Fst into t score
squared.t.scores = Fst*(n-2)/(1-Fst)

#### recalibrated p-values
gif = 25
adj.p.values = pf( squared.t.scores/gif , df1 = 2, df2 = n-3, lower = FALSE )

hist(adj.p.values,prob=TRUE)

#### Benjamini Hochberg procedure
alpha = 1e-10
L = length(adj.p.values)
# return a list of candidates with an expected FDR of alpha.
w = which(sort(adj.p.values) < alpha * (1:L) / L)
candidates = order(adj.p.values)[w]
limite = max(adj.p.values[candidates])

#### Manhattan plot
plot( 1:length(adj.p.values),-log10(adj.p.values) ,
      main = "Manhattan Plot" ,
      xlab = "indices",
      ylab="- Log P-value",
      pch=19, cex = .5)
#add limite
```

```
abline( -log10(limite), 0, col = "green", lty = 6, lwd = 3 )
```

5 Contact

If you need assistance, do not hesitate to send us an email (kevin.caye@imag.fr or olivier.francois@imag.fr).

Références