

# Package ‘tess3r’

October 14, 2015

**Type** Package

**Title** An R Package for Population Genetics Study

**Version** 0.1

**Date** 2015-08-13

**Author** Kevin CAYE

**Maintainer** kevin.caye@imag.fr

**Description** This R package implements the TESS3 program and tools useful to plot program outputs.

**License** GNU GENERAL PUBLIC LICENSE

**LazyData** TRUE

**LinkingTo** Rcpp

**Imports** Rcpp,  
fields,  
graphics,  
stats,  
RColorBrewer

**Suggests** testthat,  
knitr

**VignetteBuilder** knitr

## R topics documented:

coord . . . . .	2
createGridFromAsciiRaster . . . . .	2
getConstraintsFromAsciiRaster . . . . .	3
maps . . . . .	3
tess3 . . . . .	5
tess3Project-class . . . . .	7
tess3r . . . . .	7
<b>Index</b>	<b>8</b>

---

coord	<i>Input file for <a href="#">tess3</a></i>
-------	---

---

### Description

Description of the `coord` format. The `coord` format can be used as an input format for coordinates matrices in the functions [tess3](#).

### Details

The `geno` format has one row for each individual. Each row contains the longitude and latitude coordinates of each individual.

Here is an example of a coordinates matrix using the `coord` format with 3 individuals and 4 loci:

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

### Author(s)

Kevin Caye

---

createGridFromAsciiRaster	<i>Create grid on which coefficients will be displayed</i>
---------------------------	--

---

### Description

`createGridFromAsciiRaster` returns a grid to use to project clusters on a map. The grid is computed from an ascii raster file.

### Usage

```
createGridFromAsciiRaster(file)
```

### Arguments

file	A character string containing a the path to the ascii raster file.
------	--

### Value

`createGridFromAsciiRaster` returns a matrix which represent the grid extract from the raster file.

### See Also

This function was made to be used with [getConstraintsFromAsciiRaster](#) and [maps](#).

---

```
getConstraintsFromAsciiRaster
```

*Create grid on which coefficients will be displayed*

---

### Description

`getConstraintsFromAsciiRaster` compute a constraint matrix according to the `cell_value_min` and `cell_value_max` values.

### Usage

```
getConstraintsFromAsciiRaster(file, cell_value_min = NULL,
                              cell_value_max = NULL)
```

### Arguments

<code>file</code>	A character string containing a the path to the ascii raster file.
<code>cell_value_min</code>	All case values inferior to this numeric value will be set to FALSE on the grid.
<code>cell_value_max</code>	All case values superior to this numeric value will be set to FALSE on the grid.

### Value

`getConstraintsFromAsciiRaster` returns a matrix with boolean cells. Cells is set to TRUE if user wants to project on this map's cell

### See Also

This function was made to be used with `createGridFromAsciiRaster` and `maps`.

---

```
maps
```

*Display maps of membership/admixture coefficients.*

---

### Description

`maps` projects membership/admixture coefficients on a grid using `Krig`. Gradients in coefficients are represented by gradients in colors. if `onemap == T` & `method == "treshold"` only coefficients > 0.5 are plotted. if `onemap == T` & `method == "max"` at each point the cluster for which the coefficient is maximal is plotted (even if the value is less than 0.5). if `onemap == F` all values are plotted (since there is one cluster represented on each map, there is no overlap problem).

### Usage

```
maps(matrix, coord, grid, constraints = NULL, method = "treshold",
      colorGradientsList = lColorGradients, onemap = T, onepage = T, ...)
```

## Arguments

matrix	Matrix of values to interpolate.
coord	Matrix of corrdinate values.
grid	Matrix with grid values. Can be generated by <a href="#">createGridFromAsciiRaster</a> .
constraints	Matrix with constraints values. Can be generated by <a href="#">createGridFromAsciiRaster</a> .
method	A character string which can be "treshold" or "max".
colorGradientsList	A list of string which defines what color are used to plot the interpolation result.
onemap	A boolean value.
onepage	A boolean value.

## See Also

This function was made to used with [createGridFromAsciiRaster](#), [maps](#) and [tess3](#)

## Examples

```
## How maps can be used to plot tess3 function output ##

#####
# runs of tess3 #
#####
# Retrieve data file name
genotype.file <- system.file("extdata/Athaliana", "Athaliana.geno", package = "tess3r")
coord.file <- system.file("extdata/Athaliana", "Athaliana.coord", package = "tess3r")
# Read coordinate file
coord <- read.coord(coord.file)
n <- nrow(coord)

project <- tess3(input.file = genotype.file,
                 input.coord = coord.file,
                 K = 1:5,
                 ploidy = 1,
                 repetitions = 1,
                 entropy = TRUE,
                 percentage = 0.2,
                 project = "new")

#####
# Plot result on map#
#####

asciiFile=system.file("extdata/", "lowResEurope.asc", package = "tess3r")
grid=createGridFromAsciiRaster(asciiFile)
# To display only altitudes above 0:
constraints=getConstraintsFromAsciiRaster(asciiFile, cell_value_min=0)

maps(matrix = Q( project, K = 3, run = 1 ),
     coord = coord,
     grid=grid, constraints=constraints, method="max", main="ancestry coefficient with K = 3")
```

---

tess3	<i>Estimates individual ancestry coefficients, ancestral allele frequencies and an ancestral allele frequency differentiation statistic.</i>
-------	--

---

## Description

**tess3** estimates admixture coefficients using a graph based Non-Negative Matrix Factorization algorithms, and provide STRUCTURE-like outputs. **tess3** also computes an ancestral allele frequency differentiation statistic

## Usage

```
tess3(input.file, input.coord, K, project = "continue", repetitions = 1,
      alpha = 0.001, tolerance = 1e-05, entropy = FALSE, percentage = 0.05,
      I = 0, iterations = 200, ploidy = 1, seed = -1, CPU = 1,
      Q.input.file = "")

## S4 method for signature 'tess3Project,ANY'
plot(x, y, ...)
```

## Arguments

<code>input.file</code>	A character string containing a the path to the input genotype file, a genotypic matrix in the <a href="#">geno</a> format.
<code>input.coord</code>	A character string containing a the path to the input coordinate file, a coordinate matrix in the <a href="#">coord</a> format.
<code>K</code>	An integer vector corresponding to the number of ancestral populations for which the tess3 algorithm estimates have to be calculated.
<code>project</code>	A character string among "continue", "new", and "force". If "continue", the results are stored in the current project. If "new", the current project is removed and a new one is created to store the result. If "force", the results are stored in the current project even if the input file has been modified since the creation of the project.
<code>repetitions</code>	An integer corresponding with the number of repetitions for each value of K.
<code>alpha</code>	A numeric value corresponding to the tess3 regularization parameter. The results depend on the value of this parameter.
<code>tolerance</code>	A numeric value for the tolerance error.
<code>entropy</code>	A boolean value. If true, the cross-entropy criterion is calculated.
<code>percentage</code>	A numeric value between 0 and 1 containing the percentage of masked genotypes when computing the cross-entropy criterion. This option applies only if <code>entropy == TRUE</code> .
<code>I</code>	The number of SNPs to initialize the algorithm. It starts the algorithm with a run of snmf using a subset of nb.SNPs random SNPs. If this option is set with nb.SNPs, the number of randomly chosen SNPs is the minimum between 10000 and 10 % of all SNPs. This option can considerably speeds up snmf estimation for very large data sets.
<code>iterations</code>	An integer for the maximum number of iterations in algorithm.
<code>ploidy</code>	1 if haploid, 2 if diploid, n if n-ploid.

<code>seed</code>	A seed to initialize the random number generator. By default, the seed is randomly chosen.
<code>CPU</code>	A number of CPUs to run the parallel version of the algorithm. By default, the number of CPUs is 1.
<code>Q.input.file</code>	A character string containing a path to an initialization file for Q, the individual admixture coefficient matrix.

## Value

`tess3` returns an object of class `tess3Project`. The following methods can be applied to the object of class `tess3Project`:

<code>plot</code>	Plot the minimal cross-entropy in function of K.
<code>show</code>	Display information about the analyses.
<code>summary</code>	Summarize the analyses.
<code>Q</code>	Return the admixture coefficient matrix for the chosen run with K ancestral populations.
<code>G</code>	Return the ancestral allele frequency matrix for the chosen run with K ancestral populations.
<code>FST</code>	Return ancestral allele frequency differentiation statistic matrix for the chosen run with K ancestral populations.
<code>cross.entropy</code>	Return the cross-entropy criterion for the chosen runs with K ancestral populations.
<code>load.snmfProject(file.tess3Project)</code>	Load the file containing a <code>tess3Project</code> object and return the <code>tess3Project</code> object.
<code>remove.snmfProject(file.tess3Project)</code>	Erase a <code>tess3Project</code> object. Caution: All the files associated with the object will be removed.

## Examples

```
### Example of analyses using snmf ###
# dataset simulated from the plant species Arabidopsis thaliana
# It contains 26943 SNPs for 170 individuals.
athaliana.genofile <- system.file("extdata/Athaliana", "Athaliana.geno", package = "tess3r")
athaliana.coord <- system.file("extdata/Athaliana", "Athaliana.coord", package = "tess3r")

#####
# runs of tess3 #
#####

# main options, K: (the number of ancestral populations),
#           entropy: calculate the cross-entropy criterion,

# Runs with K between 1 and 5 with cross-entropy and 2 repetitions.
project <- tess3(athaliana.genofile, athaliana.coord, K=1:5, entropy = TRUE, repetitions = 2,
                 project = "new")

# plot cross-entropy criterion of all runs of the project
plot(project, lwd = 5, col = "red", pch=1)
```

```
# get the cross-entropy of each run for K = 4
ce = cross.entropy(project, K = 4)

# select the run with the lowest cross-entropy
best = which.min(ce)

# plot the best run for K = 3 (ancestry coefficients).
barplot(t(Q(project, K = 3, run = best)), col = c(2:4) )
```

---

tess3Project-class *An S4 class to represent a tess3 project.*

---

### Description

An S4 class to represent a tess3 project.

### See Also

You can see how to use method of this class in [tess3](#)

---

tess3r	<i>tess3r : An R Package for Population Genetics Study</i>
--------	--

---

### Description

This R package implements the TESS3 program and tools useful to plot program outputs.

# Index

## \*Topic format

`coord`, [2](#)

`coord`, [2](#), [5](#)

`createGridFromAsciiRaster`, [2](#), [3](#), [4](#)

`cross.entropy(tess3)`, [5](#)

`example_coord(coord)`, [2](#)

`FST(tess3)`, [5](#)

`G(tess3)`, [5](#)

`geno`, [5](#)

`getConstraintsFromAsciiRaster`, [2](#),  
[3](#)

`Krig`, [3](#)

`load.snmfProject(tess3)`, [5](#)

`maps`, [2](#), [3](#), [3](#), [4](#)

`plot, tess3Project, ANY-method`  
`(tess3)`, [5](#)

`Q(tess3)`, [5](#)

`remove.snmfProject(tess3)`, [5](#)

`tess3`, [2](#), [4](#), [5](#), [5](#), [7](#)

`tess3Project-class`, [7](#)

`tess3r`, [7](#)

`tess3r-package(tess3r)`, [7](#)