Package 'R.SamBada'

May 9, 2019

Type Package

Title Processing Pipeline for 'SamBada' from Pre- To Post-Processing

Version 0.1.1

Date 2019-05-07

Author Solange Duruz, Sylvie Stucki, Oliver Selmoni, Elia Vajana

Maintainer Solange Duruz <solange.duruz@alumni.epfl.ch>

Description Processing pipeline for 'SamBada' from pre- to post-processing.
'SamBada' is a landscape genomic software designed to run univariate or multivariate logistic regression between the presence of a genotype and one or several environmental variables. See Stucki (2017) <doi:10.1111/1755-0998.12629> and <https://github.com/Sylvie/sambada>.
The package provides functions that can be classified into four categories:
1) Install 'SamBada'
2) Preprocessing (prepare genomic file into standards compatible with 'SamBada' and apply quality-control; retrieve environmental conditions at sampling location; prepare environmental file including removal of correlated variables and computation of population structure)
3) Processing (run 'SamBada' on multiple cores using 'Supervision')
4) Post-processing (calculate p-values and q-values, produce interactive Manhattan plots and query 'Ensembl' database, produce maps).

License GPL (>= 2)

Imports SNPRelate, gdsfmt

LinkingTo

RoxygenNote 6.1.0

Suggests Rcpp, utils, data.table, shiny, plotly, httr, biomaRt, ggplot2, sp, packcircles, raster, mapplots, spdep, rgdal, gdalUtils, rworldmap, doParallel, foreach, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2019-05-09 10:50:02 UTC

R topics documented:

`R.SamBada-package` *R.SamBada: A package for running samBada within R with pipeline from pre to post-processing*

Description

The R.SamBada package provides functions that can be classified into four categories: Install samBada, Preprocessing, Running samBada and Post-processing.

Install samBada functions

You can download samBada (if not already on your computer) from GitHub using the function `downloadSambada`

Preprocessing functions

The Preprocessing functions contain three functions:

- `prepareGeno`: translate genomic file to samBada's input file while applying genomic filters
- `setLocation`: opens local web page with interactive map to assign sample location
- `createEnv`: create your environmental file from file location from local raster or global worldclim database
- `prepareEnv`: reduce environmental file with correlated variables and analyse population structure

Running samBada function

To run samBada, you will want to use the function: `sambadaParallel`

Postprocessing functions

The Postprocessing functions contain three functions:

- prepareOutput: calculate p and q-values from samBada output and retrieve SNP position for manhattan plots
- plotManhattan: create a manhattan plot of one or several environmental variables
- plotResultInteractive: start an interactive local web page to query a manhattan plot with maps, plots and ensembl query result
- plotMap: create a map of marker, population structure or environmental variable distribution

---

`createEnv` *Create env file from raster file(s) and/or global database present in the raster r package*

Description

Create env file as an input for SamBada (it is recommended to run prepare_env function before running samBada) raster file(s) and/or global database present in the raster r package

Usage

```
createEnv(locationFileName, outputFile, x = NULL, y = NULL,
  locationProj = NULL, separator = ",", worldclim = TRUE,
  srtm = FALSE, saveDownload, rasterName = NULL, rasterProj = NULL,
  directory = FALSE, interactiveChecks, verbose = TRUE)
```

Arguments

`locationFileName`	char Name of the file containing location of individuals. Must be in the active directory. Supported extension are .csv, .shp. All columns present in this file will also be present in the output file
`outputFile`	char Name of the output file. Must have a .csv extension.
`x`	char Name of the x (or longitude if not projected coordinate system) column in the `locationFileName`. Required if `locationFileName` extension is .csv
`y`	char Name of the y (or latitude if not projected coordinate system) column in the `locationFileName`. Required if `locationFileName` extension is .csv
`locationProj`	integer Coordinate system EPSG code of the `locationFileName`. If `locationFileName` is already georeferenced, this argument will be skipped. Required if `locationFileName` extension is csv.
`separator`	char The separator used to separate columns in your `locationFileName`
`worldclim`	logical If TRUE worldclim bio, tmin, tmax and prec variables will be downloaded at a resolution of 0.5 minutes of degree (the finest resolution). Rely rgdal and gdalUtils R package to merge the tiles. The downloaded tiles will be stored in the (new) wc0.5 directory of the active directory

srtm	logical If TRUE the SRTM (altitude) variables will be downloaded at a resolution ... Rely rgdal and gdalUtils R package to merge the tiles. The downloaded tiles will be stored in the (new) wc0.5 directory of the active directory
saveDownload	logical If TRUE (and if wordclim or srtm is TRUE), the tiles downloaded from global databases will be saved in a non-temporary directory. We recommend setting this parameter to true so that rasters can be used later (post-processing). If wordclim and srtm are FALSE, either value (TRUE/FALSE) will have no effect
rasterName	char or list Name or list of name of raster files to import. Supported format are the one of raster package. If directory is TRUE then the path to the directory. Can be set to null if worldclim or srtm are set to TRUE.
rasterProj	integer or list of integer Coordinate system EPSG code of the rasterlayer. If rasterlayer is already georeferenced, this argument will be skipped. If rasterName is a list, can be either a single number if all projections are the same or a list of projection for all files if different. If directory is TRUE, can only contain one number (all projections must be equal or rasters must be georeferenced)
directory	logical If true, all .tif, .gtiff, .img, .sdat, . present in rasterName will be loaded
interactiveChecks	logical If TRUE, shows loaded rasters and point locations
verbose	logical If TRUE, indication on process will be shown

Details

If you set worldclim=TRUE, then tmin10 represents the minimum temperature in october. Similarly tmax, tavg and prec refers to maximum temperature, average temperature and precipitation. The bio1-bio19 are bioclim variables are computed from these indices and are described here. Temperature are given in 10 degree C and precipitation in mm. The funciton always downloads the best resolution available (30 seconds for worldclim dataset and 90m for SRTM). This function requires that you define the EPSG code of your projection system. If you work with lat/long global projection, then you most probably work with WGS 84 whose EPSG is 4326.

Value

None

Author(s)

Solange Duruz

Examples

```
## Not run:
# Worldclim download only with sample data from R.SamBada
createEnv(locationFileName=system.file("extdata", "uganda-subset.csv", package = "R.SamBada"),
     outputFile=file.path(tempdir(),'uganda-subset-env.csv'), x='longitude',y='latitude',
      locationProj=4326, worldclim=TRUE,saveDownload=FALSE,interactiveChecks=TRUE)

# Own raster (fictitious examples) + worldclim download
```

```
createEnv(rasterName=c('prec.tif','tmin.sdat'),locationFileName='MyFile.shp',
      outputFile='MyFile-env.csv', rasterProj=c(4326,21781), worldclim=TRUE,
      saveDownload=TRUE,interactiveChecks=TRUE)

## End(Not run)
```

`downloadSambada` *Download samBada*

Description

Downloads from GitHub the version of samBada that corresponds to your OS. Unzips the folder and adds the path to the binary folder to the environmental path variable. This operation is only valid for the current R session. You must run change_path for every new R session. Alternatively, you can manually edit your "PATH" environmental variable permanently on your OS so that it entails the path to the binaries folder of sambada (this procedure different for every OS).

Usage

```
downloadSambada(directory = NULL)
```

Arguments

`directory`	character The directory where sambada should be downloaded. If null, downloads in a (new) folder named sambada in the active directory.

Author(s)

Solange Duruz

Examples

```
# Downloads SamBada to temporary folder (tempdir)
downloadSambada(tempdir())
```

`plotManhattan` *Manhattan plot*

Description

Plot the manhattan plot for a given environmental data

Usage

```
plotManhattan(preparedOutput, varEnv, valueName, chromo = "all",
  saveType = NULL, threshold = NULL, highlight = NULL)
```

Arguments

preparedOutput	char The prepared output list from prepare_output function
varEnv	char The name of the environmental variable one wish to study. Can be a vector of char if you want to plot several varEnv at a row. If saveType is NULL, the program prompts to continue. If saveType is png or pdf, several files are saved
valueName	char Name of the p- or q-value one wish to plot the manhattan on. This can be either pvalueG, pvalueW, qvalueG, qvalueW for G- or Waldscore respectively.
chromo	char/integer Name or vector of name of the chromosome to investigate. If all is chosen (default), all numerical chromosome will be mapped. If your sambada output is large (typically if you are working with more than 50K genomic file), you should probably map a subset of your dataset (e.g. chromo=1)
saveType	char One of NULL, 'png' or 'pdf'. If NULL is set, the plot will be shown in the R plotting window. Otherwise, it will be saved in the specified format in your working directory with the name 'manhattan-' followed by varEnv.
threshold	double A digit number indicating a value to draw a threshold line
highlight	char Name of the genotype to highlight in red on plot (should be SNPName_Genotype e.g. 'ARS-BFGL-NGS-106879_AA')

Value

The last plot object (if several varEnv are specified, only the last one is returned)

Author(s)

Solange Duruz

Examples

```
# Example with data from the package
# First copy needed files into the temporary directory
file.copy(system.file("extdata", "uganda-subset-mol-Out-2.csv", package = "R.SamBada"),
    file.path(tempdir(),'uganda-subset-mol-Out-2.csv'), overwrite=TRUE)
file.copy(system.file("extdata", "uganda-subset-mol-storey.csv", package = "R.SamBada"),
    file.path(tempdir(),'uganda-subset-mol-storey.csv'), overwrite=TRUE)
if(Sys.info()['sysname']=='Windows'){
  file.copy(system.file("extdata", "uganda-subset-mol_windows.gds", package = "R.SamBada"),
      file.path(tempdir(),'uganda-subset-mol.gds'), overwrite=TRUE) #If you run Windows
} else {
  file.copy(system.file("extdata", "uganda-subset-mol_unix.gds", package = "R.SamBada"),
      file.path(tempdir(),'uganda-subset-mol.gds'), overwrite=TRUE) #If you run Unix
}
# Run prepareOutput
prep=prepareOutput(file.path(tempdir(),'uganda-subset-mol'),2,popStr=TRUE,
     interactiveChecks=FALSE)
####################
# Run plotManhattan
####################
plotManhattan(prep, c('bio1'),chromo='all',valueName='pvalueG')
```

```
# Example with several environmental variables
plotManhattan(prep,c('bio1','bio2'),'pvalueG')
```

---

plotMap *Plotting of maps*

Description

Plots several kinds of maps (environmental variable distribution, population structure, marker absence or presence, autocorrelation of marker). Unlike plotResultInteractive, the resulting maps are non-interactive. The function can handle several marker/variables at once and create separate output files.

Usage

```
plotMap(envFile, x, y, locationProj, popStrCol, gdsFile, markerName,
  mapType, varEnvName, SAMethod = NULL, SAThreshold = NULL,
  saveType = NULL, rasterName = NULL, simultaneous = FALSE)
```

Arguments

envFile	char The file containing the input environmental variable of sambada.
x	char The name of the column corresponding to the x-coordinate in the envFile. Can be set to null if unknown, in this case the maps will not be available
y	char The name of the column corresponding to the y-coordinate in the env file. Can be set to null if x is null.
locationProj	integer EPSG code of the geographical projection in the envFile
popStrCol	char The name or vector of name of column(s) in envFile describing population structure. If provided, additional layers on the map will be available representing population structure.
gdsFile	char The GDS file created in the preprocessing of sambada. If null, will try with envFile(without -env.csv) and .gds
markerName	name of the marker to be plotter if mapType is 'marker' or 'AS'. markerName can be found in preparedOutput$sambadaOutput[,"] where preparedOutput would be the result of the function prepareOutput
mapType	char A string or vector of string containing one or several of 'marker' (presence/absence of marker), 'env' (environmental variable distribution), 'popStr' (population variable on continuous scale), 'popPieChart' (belonging to a population in pie charts), 'AS' (autocorrelation of the marker). Note that the background of all maps, if found, will be the raster of the environmental variable. Thus the 'env' mapType is preferred when no raster is provided. For the 'AS' type, it is calculated on the fly for the markers provided and not the one possibly calculated by sambada.

varEnvName	char Name of the environmental variable. If a raster of the variable is located in your working directory, you can provide varEnvName even for mapType such as 'marker' or 'AS'. The function will scan the folder of your working directory for raster with the same name as varEnvName (and commonly used extension for raster) and put it as background.
SAMethod	char If mapType contains 'AS', then you must specify the method for setting the weights of neighbours. Can be one of 'knn' (k-nearest neighbours) or 'distance'
SAThreshold	char If mapType contains 'AS' and SAMethod is 'knn' then the number of neighbours. If SAMethod is 'distance' then the distance in map-unit (unless you use a spherical projection (latitude/longitude), in which case you should use km)
saveType	char One of NULL, 'png' or 'pdf'. If NULL is set, the maps will be shown in the R plotting window. Otherwise, it will be saved in the specified format in your working directory.
rasterName	char If a raster file with the environmental variable distribution exists with a different name than varEnvName, provide it here (including extension)
simultaneous	boolean If TRUE and mapType contains several kinds of maps, all maps corresponding to the same marker will be plotted on the same window. The resulting maps can be very small.

Value

None

Author(s)

Solange Duruz

Examples

```
# Define right GDS file according to your OS
if(Sys.info()['sysname']=='Windows'){
  gdsFile=system.file("extdata", "uganda-subset-mol_windows.gds", package = "R.SamBada")
} else {
  gdsFile=system.file("extdata", "uganda-subset-mol_unix.gds", package = "R.SamBada")
}
#############
# Run plotMap
#############
plotMap(envFile=system.file("extdata", "uganda-subset-env-export.csv", package = "R.SamBada"),
     x='longitude', y='latitude', locationProj=4326,  popStrCol='pop1', gdsFile=gdsFile,
     markerName='Hapmap28985-BTA-73836_GG', mapType='marker', varEnvName='bio1',
     simultaneous=FALSE)

# Maps of marker and population structure (two subplot)
plotMap(envFile=system.file("extdata", "uganda-subset-env-export.csv", package = "R.SamBada"),
     'longitude','latitude', locationProj=4326,  popStrCol='pop1',
     gdsFile=gdsFile, markerName='Hapmap28985-BTA-73836_GG',
     mapType=c('marker', 'popStr'), varEnvName='bio1', simultaneous=TRUE)
```

plotResultInteractive *Interactive plotting of results*

Description

Plots the manhattan plot for a given environmental variable. The plot is interactive and a map of the distribution of the marker can be retrieved as well as nearby genes listed in Ensembl.

Usage

```
plotResultInteractive(preparedOutput, varEnv, envFile, species = NULL,
  pass = NULL, x = NULL, y = NULL, valueName = "pvalueG",
  chromo = "all", gdsFile = NULL, IDCol = NULL, popStrCol = NULL)
```

Arguments

preparedOutput	char The prepared output list from prepare_output function
varEnv	char The name of the environmental variable one wish to study (as in the header of envFile)
envFile	char The file containing the input environmental variable of sambada.
species	char The abbreviated latin name of the species without capitals nor punctuation (e.g. btaurus, chircus,...). Can be set to null if species not present in ensembl database
pass	integer Number of BP around a SNP in which to look for an annotation in Ensembl. Set to null if species is null
x	char The name of the column corresponding to the x-coordinate in the envFile. Can be set to null if unknown, in this case the maps will not be available
y	char The name of the column corresponding to the y-coordinate in the env file. Can be set to null if x is null.
valueName	char Name of the p- or q-value one wish to plot the manhattan on. This can be either pvalueG, pvalueW, qvalueG, qvalueW for G- or Waldscore respectively.
chromo	char/integer Name or vector of name of the chromosome to investigate. If all is chosen (default), all numerical chromosome will be mapped. If your sambada output is large (typically if you are working with more than 50K genomic file), you should probably map a subset of your dataset (e.g. chromo=1)
gdsFile	char The GDS file created in the preprocessing of sambada. If null, will try with envFile(without -env.csv or -env-export.csv) and .gds
IDCol	char The name of the column in envFile corresponding to the ID of the individual. If provided, hover on the output map will give the id of the animal
popStrCol	char The name or vector of name of column(s) in envFile describing population structure. If provided, additional layers on the map will be available representing population structure.

Value

None

Author(s)

Solange Duruz

Examples

```
## Not run:
# Example with data from the package
# First copy needed files into the temporary directory
file.copy(system.file("extdata", "uganda-subset-mol-Out-2.csv", package = "R.SamBada"),
     file.path(tempdir(),'uganda-subset-mol-Out-2.csv'), overwrite=TRUE)
file.copy(system.file("extdata", "uganda-subset-mol-storey.csv", package = "R.SamBada"),
     file.path(tempdir(),'uganda-subset-mol-storey.csv'), overwrite=TRUE)
file.copy(system.file("extdata", "uganda-subset-env-export.csv", package = "R.SamBada"),
     file.path(tempdir(),'uganda-subset-env-export.csv'), overwrite=TRUE)
if(Sys.info()['sysname']=='Windows'){
  file.copy(system.file("extdata", "uganda-subset-mol_windows.gds", package = "R.SamBada"),
       file.path(tempdir(),'uganda-subset-mol.gds'), overwrite=TRUE) #If you run Windows
} else {
  file.copy(system.file("extdata", "uganda-subset-mol_unix.gds", package = "R.SamBada"),
       file.path(tempdir(),'uganda-subset-mol.gds'), overwrite=TRUE)
}
# Run prepareOutput
prep=prepareOutput(file.path(tempdir(),'uganda-subset-mol'),2,popStr=TRUE,
     interactiveChecks=FALSE)
####################
# Run plotResultInteractive
####################
plotResultInteractive(prep,'bio1','uganda-subset-env-export.csv',species='btaurus',
     pass=25000,x='longitude',y='latitude', gdsFile='uganda-subset-mol.gds',
     IDCol='short_name',popStrCol='pop1')

## End(Not run)
```

---

prepareEnv *Prepare environmental input*

Description

Writes a new environmental file that sambada can work with after having removed too correlated variables. Also calculates population structure from a PCA in SNPRelate and add it at the end of the environmental file

Usage

```
prepareEnv(envFile, outputFile, maxCorr, idName, separator = " ",
  genoFile = NULL, numPc = 0.5, mafThresh = NULL,
  missingnessThresh = NULL, ldThresh = NULL, numPop = -1,
  clustMethod = "kmeans", includeCol = NULL, excludeCol = NULL,
  popStrCol = NULL, x, y, locationProj, interactiveChecks = FALSE,
  verbose = TRUE)
```

Arguments

envFile	char Name of the input environmental file (must be in active directory). Can be .csv or .shp
outputFile	char Name of the output file. Must have a .csv extension.
maxCorr	double A number between 0 and 1 specifying the maximum allowable correlation coefficient between environmental files. If above, one of the variables will be deleted
idName	char Name of the id in the environmental file matching the one of genoFile
separator	char If envFile is .csv, the separator character. If file created with create_env, separator is ' '
genoFile	char (optional) Name of the input genomic file (must be in active directory). If not null, population variable will be calculated from a PCA relying on the SNPRelate package. Can be .gds, .ped, .bed, .vcf. If different from .gds, a gds file (SNPrelate specific format) will be created
numPc	double If above 1, number of principal components to analyze. If between 0 and 1, automatic detection of number of PC (the program will find the first leap in the proportion of variance where the ratio (difference in variance between PC x and x+1)/(variance of PC x) is greater than numPc. If 0, PCA and population structure will not be computed: in that case, the genoFile will only be used to make the sample order in the envFile match the one of the genoFile (necessary for sambada's computation). Set it to 0 if genoFile is null
mafThresh	double A number between 0 and 1 specifying the Major Allele Frequency (MAF) filtering when computing PCA (if null no filtering on MAF will be computed)
missingnessThresh	double A number between 0 and 1 specifying the missing rate filtering when computing PCS(if null no filtering on missing rate will be computed)
ldThresh	double A number between 0 and 1 specifying the linkage disequilibrium (LD) rate filtering before computing the PCA (if null no filtering on LD will be computed)
numPop	integer If not null, clustering based on numPc first PC will be computed to divide into numPop populations. If -1 automatic detection of number of cluster (elbow method if clustMethod = 'kmeans', maximise branch length if clustMethod = 'hclust'). If null, no clustering will be computed: if genoFile is set, principal component scores will be included as population information in the final file.
clustMethod	char One of 'kmeans' or 'hclust' for K-means and hierarchical clustering respectively. Default 'kmeans'

includeCol — character vector Columns in the environmental file to be considered as variables. If none specified, all numeric variables will be considered as env var except for the id

excludeCol — character vector Columns in the environmental file to exclude in the output (non-variable column). If none specified, all numeric variables will be considered as environmental variables except for the id

popStrCol — character vector Columns in the environmental file describing population structure (ran elsewhere). Those columns won't be excluded when correlated with environmental files

x — character Name of the column corresponding to the x coordinate (or longitude if spherical coordinate). If not null, x column won't be removed even if correlated with other variable. This parameter is also used to display the map of the population structure.

y — character Name of the column corresponding to the y coordinate (or latitude if spherical coordinate). If not null, y column won't be removed even if correlated with other variable. This parameter is also used to display the map of the population structure.

locationProj — integer EPSG code of the projection of x-y coordinate

interactiveChecks — logical If TRUE, plots will show up showing number of populations chosen, and correlation between variables and the user can interactively change the chosen threshold for maxCorr and numPop (optional, default value=FALSE)

verbose — boolean If true show information about progress of the process

Details

The population structure is calculated as a PCA of all the SNPs that pass the filtering (maf, ld, missingness). You can either choose to use the score of the X first components to evaluate the population structure (set 'numPop' to NULL) or you can compute a "membership coefficient" to a cluster of individuals based on the scores on the first X components. You can choose between two clustering algorithm (k-means or hierarchical cluster in the 'clustMethod' argument). One of the option to decide the number of PCs that you should keep is to detect a bump in the proportion of variance explained and keep all the PC before the bump.

Value

None

Author(s)

Solange Duruz, Oliver Selmoni

Examples

```
################
# Run prepareEnv
################
#Without calculating population structure.
```

```
prepareEnv(envFile=system.file("extdata", "uganda-subset-env.csv", package = "R.SamBada"),
     outputFile=file.path(tempdir(),'uganda-subset-env-export.csv'), maxCorr=0.8,
     numPc=0, idName='short_name', x='longitude',y='latitude', locationProj=4326,
     interactiveChecks = FALSE)

# While it is not mandatory to provide gdsFile, it is recommended to define it so that IDs
# in envrionmental and genomic file are in the same order (gdsFile also needed to compute
# population structure)

# determine gdsFile according to OS
if(Sys.info()['sysname']=='Windows'){
  gdsFile="uganda-subset-mol_windows.gds"
} else {
  gdsFile="uganda-subset-mol_unix.gds"
}

#Calculating PCA-based population structure
prepareEnv(envFile=system.file("extdata", "uganda-subset-env.csv", package = "R.SamBada"),
     outputFile=file.path(tempdir(),'uganda-subset-env-export.csv'), maxCorr=0.8,
    idName='short_name', genoFile=system.file("extdata", gdsFile, package = "R.SamBada"),
     numPc=0.2, mafThresh=0.05, missingnessThresh=0.1, ldThresh=0.2, numPop=NULL,
     x='longitude', y='latitude', locationProj=4326, interactiveChecks = TRUE)

#Calculating structure membership coefficient based on kmeans clustering
prepareEnv(envFile=system.file("extdata", "uganda-subset-env.csv", package = "R.SamBada"),
     outputFile=file.path(tempdir(),'uganda-subset-env-export.csv'), maxCorr=0.8,
    idName='short_name', genoFile=system.file("extdata", gdsFile, package = "R.SamBada"),
     numPc=0.2, mafThresh=0.05, missingnessThresh=0.1, ldThresh=0.2, numPop=NULL,
     x='longitude', y='latitude', locationProj=4326, interactiveChecks = TRUE)
```

`prepareGeno` *Prepare genomic input*

Description

Writes a new genomic file that sambada can work with after having applied the selected genomic filtering options. The output file has the same name as the input file but with a .csv extension

Usage

```
prepareGeno(fileName, outputFile, saveGDS, mafThresh = NULL,
  missingnessThresh = NULL, ldThresh = NULL, mgfThresh = NULL,
  directory = NULL, interactiveChecks = FALSE, verbose = FALSE)
```

Arguments

`fileName`	char Name of the input file (must be in active directory). Can be .gds, .ped, .bed, .vcf. If different from .gds, a gds file (SNPrelate specific format) will be created unless no filtering options are chosen

outputFile	char Name of the output file. Must be a .csv
saveGDS	logical If true (and if the input file extension is different from GDS) the GDS file will be saved. We recommend to set this parameter to TRUE to save time in subsequent functions that rely on GDS file
mafThresh	double A number between 0 and 1 specifying the Major Allele Frequency (MAF) filtering (if null no filtering on MAF will be computed)
missingnessThresh	double A number between 0 and 1 specifying the missing rate filtering (if null no filtering on missing rate will be computed)
ldThresh	double A number between 0 and 1 specifying the linkage disequilibrium (LD) rate filtering (if null no filtering on LD will be computed)
mgfThresh	double A number between 0 and 1 specifying the Major Genotype Frequency (MGF) rate filtering (if null no filtering on MGF will be computed). NB: sambada computations rely on genotypes
directory	char The directory where binaries of sambada are saved. This parameter is not necessary if directory path is permanently stored in the PATH environmental variable or if a function invoking sambada executable (prepareGeno or sambadaParallel) has been already run in the R active session.
interactiveChecks	logical If TRUE, plots will show up showing distribution of allele frequency etc... and the user can interactively change the chosen threshold for mafThresh, missingnessThresh, mgfThresh (optional, default value=FALSE)
verbose	logical Turn on verbose mode

Value

None

Author(s)

Solange Duruz, Oliver Selmoni

Examples

```
# Example with data from the package
# You first need to download sambada and add the directory input parameter to specify where
# you saved it, unless you add it to your PATH environmental varialbe
#################
# Run prepareGeno
#################
# Example with ped input file, no filtering
prepareGeno(system.file("extdata", "uganda-subset-mol.ped", package = "R.SamBada"),
    outputFile=file.path(tempdir(),'/uganda-subset-mol.csv'),FALSE, interactiveChecks=FALSE)


# Example with gds file and filtering
# Define right GDS file according to your OS
if(Sys.info()['sysname']=='Windows'){
```

```
  gdsFile=system.file("extdata", "uganda-subset-mol_windows.gds", package = "R.SamBada")
} else {
  gdsFile=system.file("extdata", "uganda-subset-mol_unix.gds", package = "R.SamBada")
}
prepareGeno(gdsFile, outputFile=file.path(tempdir(),'/uganda-subset-mol.csv'),
     saveGDS=FALSE,mafThresh=0.05, missingnessThresh=0.1,interactiveChecks=FALSE)

# Run prepareGeno with interactiveChecks=TRUE
prepareGeno(fileName=system.file("extdata", "uganda-subset-mol.ped", package = "R.SamBada"),
     outputFile=file.path(tempdir(),'/uganda-subset-mol.csv'),TRUE, mafThresh=0.05,
     missingnessThresh=0.05, mgfThresh=0.8,interactiveChecks=TRUE)
```

`prepareOutput` *Prepare output (useful for all postprocessing analysis)*

Description

Read sambada's output and prepare it by retrieving the snp position and chromosome (useful for plotting manhattan)

Usage

```
prepareOutput(sambadaname, dimMax, gdsFile = NULL, popStr = FALSE,
  nrows = NULL, interactiveChecks = TRUE)
```

Arguments

`sambadaname`	char The name of the genofile without extension name given to sambada (or outputfile of sambada without the ending -Out-Dim.csv)
`dimMax`	integer The maximum number of dimension given in sambada
`gdsFile`	char Name of the gds file associated with sambada's input file. If null, will try with `sambadaname.gds`
`popStr`	logical Indicates whether sambada was run using the POPSTRVAR parameter (i.e. population structure was taken into account). Default false
`nrows`	integer Specifies the number of line to read from the input file. Useful if `saveType` 'END ALL' was used in `sambadaParallel` and that the number of models run is large so that the reading and processing is too slow. The `saveType` 'END' parameter ensures that most significant models are located at the top of the file.
`interactiveChecks`	logical If TRUE, plots showing the distribution of p-values and estimates of pi0 (to adjust q-values) will be drawn

Value

a list containing a) `$sambadaOutput` a matrix containing the output from sambada with 3 additional column: corresponding snp, chromosome and position of the marker b) `$chrSNPNum` The total number of SNPs in each chromosome c) `$chrMaxPos` The highest position found in each chromosome

Author(s)

Solange Duruz, Sylvie Stucki

Examples

```
# Example with data from the package
# First copy needed files into the temporary directory
file.copy(system.file("extdata", "uganda-subset-mol-Out-2.csv", package = "R.SamBada"),
     file.path(tempdir(),'uganda-subset-mol-Out-2.csv'), overwrite=TRUE)
file.copy(system.file("extdata", "uganda-subset-mol-storey.csv", package = "R.SamBada"),
     file.path(tempdir(),'uganda-subset-mol-storey.csv'), overwrite=TRUE)
if(Sys.info()['sysname']=='Windows'){
  file.copy(system.file("extdata", "uganda-subset-mol_windows.gds", package = "R.SamBada"),
       file.path(tempdir(),'uganda-subset-mol.gds'), overwrite=TRUE) #If you run Windows
} else {
  file.copy(system.file("extdata", "uganda-subset-mol_unix.gds", package = "R.SamBada"),
       file.path(tempdir(),'uganda-subset-mol.gds'), overwrite=TRUE)
}
###################
# Run prepareOutput
###################
prep=prepareOutput(file.path(tempdir(),'uganda-subset-mol'),2,popStr=TRUE,
     interactiveChecks=FALSE)
```

---

sambadaParallel *Run sambada on parallel cores*

Description

Read samBada's input file to retrieve necessary information (number of individuals etc...), split the dataset using SamBada's Supervision tool, run sambada on the splitted dataset and merge all using Supervision. This function produces the following output files: `outputFile`-Out-0.csv to `outputFile`-Out-`dimMax`.csv as well as outputFile-storey.csv (`outputFile` and `dimMax` are parameters of the function). See sambada's documentation for more information. In case you have to specify several words in one parameter, you can either specify them in one string and separate them with a space or add a vector string

Usage

```
sambadaParallel(genoFile, envFile, idGeno, idEnv, outputFile, dimMax = 1,
  cores = NULL, wordDelim = " ", saveType = "END BEST 0.05",
  populationVar = NULL, spatial = NULL, autoCorr = NULL,
  shapeFile = NULL, colSupEnv = NULL, colSupMark = NULL,
  subsetVarEnv = NULL, subsetVarMark = NULL, headers = TRUE,
  directory = NULL, keepAllFiles = FALSE)
```

Arguments

genoFile	The name of the file in the current directory of genetic information, compliant with samBada's format (use prepareGeno to transform it)
envFile	The name of the file in the current directory of environmental information (use link{createEnv} to create it and link{prepareEnv} to reduce the correlated dataset and check order)
idGeno	Name of the column in the genoFile corresponding to the id of the animals
idEnv	Name of the column in the envFile corresponding to the id of the animals
outputFile	char Base name(s) for the results file(s). Output files will be created from the base name with suffixes (e.g. -Out-)
dimMax	Maximum number of environmental variables included in the logistic models. Use 1 for univariate models, 2 for univariate and bivariates models
cores	Number of cores to use. If NULL, the #cores-1 will be used where #cores corresponds to all cores available on your computer.
wordDelim	char Word delimiter of input file(s). Default ' ',
saveType	composed of three words 1) one of 'end' or 'real' to save the result during the analysis or at the end (allows sorting of result) 2) one of 'all' or 'best' to save all models or only significant models 3) If 'best' specify the threshold of significance (before applying Bonferroni's correction). Default 'END BEST 0.05',
populationVar	one of 'first' or 'last'. This option indicates whether any explanatory variables represent the population structure. If present, the said population variables must be gathered in the input file, either on the left or on the right side of the group of environmental variables. Default null.
spatial	composed of 5 words 1) Column name (or number) for longitude 2) Column name (or number) for latitude 3) one of 'spherical' or 'cartesian': to indicate the type of coordinate 4) one of 'distance', 'gaussian', bisquare' or 'nearest': type of weighting scheme (see sambadoc) 4) Number bandwidth of weighting function Input type is (double). Units are in [m] for spherical coordinates; for cartesian coordinates, units match those of the samples' positions. Case nearest: Input type is (int)
autoCorr	composed of 3 words. 1) one of global, local or both: to indicate the type of spatial autocorrelation to compute. 2) one of env, mark or both: to indicate the variables on which to compute the analysis 3) integer The number of permutation to compute the pseudo p-value. Ex 'global both 999'
shapeFile	one of yes or no. With this option, the LISA are saved as a shapefile (in addition to the usual output)
colSupEnv	char or vector of char Name(s) of the column(s) in the environmental data to be excluded from the analysis. Default NULL
colSupMark	char or vector of char Name(s) of the column(s) in the molecular data to be excluded from the analysis. Default NULL
subsetVarEnv	char or vector of char Name(s) of the column(s) in the environmental data to be included in the analysis while the other columns are set as inactive. Default NULL

subsetVarMark	char or vector of char Name(s) of the column(s) in the molecular data to be included in the analysis while the other columns are set as inactive. Default NULL
headers	logical Presence or absence of variable names in input files Default TRUE
directory	char The directory where binaries of sambada are saved. This parameter is not necessary if directory path is permanently stored in the PATH environmental variable or if a function invoking sambada executable (prepareGeno or sambadaParallel) has been already run in the R active session.
keepAllFiles	logical If TRUE, all parameter files and split genoFile and log-files are not removed. Default FALSE

Author(s)

Solange Duruz, Sylvie Stucki

Examples

```
# Example with data from the package
# You first need to download sambada with downloadSambada(tempdir())
# Example without population structure, using only one core
sambadaParallel(genoFile=system.file("extdata", "uganda-subset-mol.csv", package = "R.SamBada"),
    envFile=system.file("extdata", "uganda-subset-env-export.csv", package = "R.SamBada"),
     idGeno='ID_indiv', idEnv='short_name', dimMax=1, cores=1, saveType='END ALL',
     outputFile=file.path(tempdir(),'uganda-subset-mol'))

# Example with population structure, using multiple core
sambadaParallel(genoFile=system.file("extdata", "uganda-subset-mol.csv", package = "R.SamBada"),
    envFile=system.file("extdata", "uganda-subset-env-export.csv", package = "R.SamBada"),
     idGeno='ID_indiv', idEnv='short_name', dimMax=2, cores=2, saveType='END ALL',
     populationVar='LAST', outputFile=file.path(tempdir(),'uganda-subset-mol'))
```

---

setLocation *Set the location of samples through a local web-application with interactive map*

Description

Helps the user defining the location of samples by opening a local web page. If the html fails to open, one must open georeftool.html manually in any web browser: the file is located within the extdata folder of the package. Once opened, the user must upload a file with at least one column corresponding to sample IDs. He can then specify the name of the column corresponding to lat/long if present. For samples without location, he must select the individuals on the list shown and click on a point of the map. The location of the map will be assigned to the chosen samples. When finished, the new file can be downloaded.

Usage

```
setLocation()
```

Author(s)

Oliver Selmoni, Solange Duruz

Examples

```
## Not run:
setLocation()

## End(Not run)
```

Index