Package ‘dartR.captive’

March 17, 2026

Type Package

Title Analysing 'SNP' Data to Support Captive Breeding

Version 1.2.2

Revision Elastic Elapid

Date 2026-02-20

Description Functions are provided that facilitate the analysis of SNP (single nucleotide polymorphism) data to answer questions regarding captive breeding and relatedness between individuals. 'dartR.captive' is part of the 'dartRverse' suit of packages.
Gruber et al. (2018) <doi:10.1111/1755-0998.12745>.
Mijangos et al. (2022) <doi:10.1111/2041-210X.13918>.

Encoding UTF-8

Depends R (>= 3.5), dartR.base, dartR.data, dartR.sim

Imports adegenet (>= 2.0.0), methods, utils, crayon, ggplot2, patchwork, stringr, data.table, gridExtra, magrittr,reshape2,tidyr,digest

Suggests SIBER, gplots, fields, igraph, rrBLUP, scales, spelling, tidyverse

License GPL (>= 3)

RoxygenNote 7.3.3

NeedsCompilation no

Author Bernd Gruber [aut, cre],
Arthur Georges [aut],
Jose L. Mijangos [aut],
Carlo Pacioni [aut],
Peter J. Unmack [ctb],
Oliver Berry [ctb],
Lindsay V. Clark [ctb],
Floriaan Devloo-Delva [ctb],
Eric Archer [ctb],
Sam Amini [ctb],
Ethan Halford [ctb]

URL https://green-striped-gecko.github.io/dartR/

BugReports https://groups.google.com/g/dartr?pli=1

Language en-US

Maintainer Bernd Gruber <bernd.gruber@canberra.edu.au>

Repository CRAN

Date/Publication 2026-03-17 08:10:19 UTC

Contents

`gl.assign.grm` *Population assignment using grm*

Description

This function takes one individual and estimates their probability of coming from individual populations from multilocus genotype frequencies.

Usage

```
gl.assign.grm(x, unknown, verbose = NULL)
```

Arguments

`x` Name of the genlight object containing the SNP data [required].

`unknown` Name of the individual to be assigned to a population [required].

`verbose` Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2, unless specified using gl.set.verbosity].

Details

This function is a re-implementation of the function multilocus_assignment from package gstudio. Description of the method used in this function can be found at: https://dyerlab.github.io/applied_population_genetics/populat
assignment.html

Value

A `data.frame` consisting of assignment probabilities for each population.

Author(s)

Custodian: Luis Mijangos – Post to https://groups.google.com/d/forum/dartr

Examples

```
require("dartR.data")
if ((requireNamespace("rrBLUP", quietly = TRUE)) & (requireNamespace("gplots", quietly = TRUE))) {
  if (isTRUE(getOption("dartR_fbm"))) platypus.gl <- gl.gen2fbm(platypus.gl)
  res <- gl.assign.grm(platypus.gl, unknown = "T27")
}
```

`gl.assign.mahal` *Calculate probabilities of assignment of an individual of unknown provenance to population based on Mahalanobis Distance*

Description

This script assigns an individual of unknown provenance to one or more target populations based on the unknown individual's proximity to population centroids; proximity is estimated using Mahalanobis Distance and a z score and probability of assignment is calculated.

The following process is followed:

1. An ordination is undertaken on the populations to again yield a series of orthogonal (independent) axes.
2. A workable subset of dimensions is chosen, that specified as dim.limit, or the number of dimensions with substantive eigenvalues (Kaiser-Guttman criterion), whichever is the smaller.
3. The Mahalobalis Distance is calculated for the unknown against each population and probability of membership of each population is calculated. The assignment probabilities are listed in support of a decision.

Usage

```
gl.assign.mahal(
  x,
  nmin = 10,
  dim.limit = NULL,
  plevel = 0.001,
  n.best = NULL,
  unknown,
  verbose = NULL
)
```

Arguments

`x`	Name of the input genlight object [required].
`nmin`	Minimum sample size for a target population to be included in the analysis [default 10].
`dim.limit`	Maximum number of dimensions to consider for the confidence ellipses [default nPop(x)-1]
`plevel`	Probability level for bounding ellipses [default 0.001].
`n.best`	If given a value, dictates the best n=n.best populations to retain for consideration (or more if their are ties). If not specified, then the putative source populations identified as possibilities by the PCA are retained. [default NULL].
`unknown`	Identity label of the focal individual whose provenance is unknown [required].
`verbose`	Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2 or as specified using gl.set.verbosity].

Details

There are three considerations to assignment. First, consider only those populations for which the unknown has no private alleles. Private alleles are an indication that the unknown does not belong to a target population (provided that the sample size is adequate, say >=10). This can be evaluated with gl.assign.pa().

A next step is to consider the PCoA plot for populations remaining after step 1. The position of the unknown in relation to the confidence ellipses is plotted by this script as a basis for narrowing down the list of putative source populations. This can be evaluated with gl.assign.pca().

The third step (delivered by this script) is to consider the assignment probabilities based on the squared Generalised Linear Distance (Mahalanobis distance) of the unknown from the centroid for each population, then to consider the probability associated with its quantile using the Chisquare approximation. In effect, this index takes into account position of the unknown in relation to the confidence envelope in all selected dimensions of the ordination. The larger the assignment probability, the greater the confidence in the assignment.

If dim.limit is set to 2, to correspond with the dimensions used in gl.assign.pa(), then the output provides a ranking of the set of putative source populations selected after the PCoA selection step.

If dim.limit is set to be > 2, then this script provides a basis for further narrowing the set of putative populations.If the unknown individual is an extreme outlier, say at less than 0.001 probability of population membership (0.999 confidence envelope), then the associated population can be eliminated from further consideration.

Warning: gl.assign.mahalanobis() treats each specified dimension equally, without regard to the percentage variation explained after ordination. If the unknown is an outlier in a lower dimension with an explanatory variance of, say, 0.1 only uses substantive dimensions from the ordination.

Each of these above approaches provides evidence, none are 100 They need to be interpreted cautiously.

In deciding the assignment, the script considers an individual to be an outlier with respect to a particular population at alpha = 0.001 as default

Value

A data frame with the results of the assignment analysis.

Author(s)

Script: Arthur Georges. Custodian: Arthur Georges – Post to https://groups.google.com/d/forum/dartr

---

`gl.assign.mahalanobis` *Calculate probabilities of assignment of an individual of unknown provenance to population based on Mahalanobis Distance*

Description

This script assigns an individual of unknown provenance to one or more target populations based on the unknown individual's proximity to population centroids; proximity is estimated using Mahalanobis Distance and a z score and probability of assignment is calculated.

The following process is followed:

1. An ordination is undertaken on the populations to again yield a series of orthogonal (independent) axes.
2. A workable subset of dimensions is chosen, that specified as dim.limit, or the number of dimensions with substantive eigenvalues (Kaiser-Guttman criterion), whichever is the smaller.
3. The Mahalobalis Distance is calculated for the unknown against each population and probability of membership of each population is calculated. The assignment probabilities are listed in support of a decision.

Usage

```
gl.assign.mahalanobis(
  x,
  nmin = 10,
  dim.limit = NULL,
  plevel = 0.001,
  n.best = NULL,
  unknown,
  verbose = NULL
)
```

Arguments

`x` Name of the input genlight object [required].

`nmin` Minimum sample size for a target population to be included in the analysis [default 10].

`dim.limit` Maximum number of dimensions to consider for the confidence ellipses [default nPop(x)-1]

`plevel` Probability level for bounding ellipses [default 0.001].

`n.best` If given a value, dictates the best n=n.best populations to retain for consideration (or more if their are ties). If not specified, then the putative source populations identified as possibilities by the PCA are retained. [default NULL].

`unknown` Identity label of the focal individual whose provenance is unknown [required].

`verbose` Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2 or as specified using gl.set.verbosity].

Details

There are three considerations to assignment. First, consider only those populations for which the unknown has no private alleles. Private alleles are an indication that the unknown does not belong to a target population (provided that the sample size is adequate, say >=10). This can be evaluated with gl.assign.pa().

A next step is to consider the PCoA plot for populations remaining after step 1. The position of the unknown in relation to the confidence ellipses is plotted by this script as a basis for narrowing down the list of putative source populations. This can be evaluated with gl.assign.pca().

The third step (delivered by this script) is to consider the assignment probabilities based on the squared Generalised Linear Distance (Mahalanobis distance) of the unknown from the centroid for each population, then to consider the probability associated with its quantile using the Chisquare approximation. In effect, this index takes into account position of the unknown in relation to the confidence envelope in all selected dimensions of the ordination. The larger the assignment probability, the greater the confidence in the assignment.

If dim.limit is set to 2, to correspond with the dimensions used in gl.assign.pa(), then the output provides a ranking of the set of putative source populations selected after the PCoA selection step.

If dim.limit is set to be > 2, then this script provides a basis for further narrowing the set of putative populations.If the unknown individual is an extreme outlier, say at less than 0.001 probability of population membership (0.999 confidence envelope), then the associated population can be eliminated from further consideration.

Warning: gl.assign.mahalanobis() treats each specified dimension equally, without regard to the percentage variation explained after ordination. If the unknown is an outlier in a lower dimension with an explanatory variance of, say, 0.1 only uses substantive dimensions from the ordination.

Each of these above approaches provides evidence, none are 100 They need to be interpreted cautiously.

In deciding the assignment, the script considers an individual to be an outlier with respect to a particular population at alpha = 0.001 as default

Value

A data frame with the results of the assignment analysis.

Author(s)

Script: Arthur Georges. Custodian: Arthur Georges – Post to https://groups.google.com/d/forum/dartr

---

gl.assign.on.genotype *Use genotype to identify populations as possible source populations for an individual of unknown provenance.*

Description

This script identifies populations for which the unknown individual has a reasonable expectation of having been drawn from those populations given its genotype and the allele frequencies in the putative source populations. The putative source populations that survive are retained and returned in a genlight object.

The algorithm computes the log-likelihood of the focal genotype under Hardy-Weinberg (HWE), then computes a Z-score and one-tailed p-value by comparing the unknown individual's log-likelihood to those from individuals in each putative source population. Significant departures from expectation renders a population unlikely to be the source for the focal unknown individual.

A suitable estimate of the expectation for the log likelihoods requires that the sample size is adequate, say >=10).

WARNING: If a putative population is not in Hardy-Weinberg equilibrium, as might occur if it includes F1 hybrids and backcrosses, then the standard deviation for the expectation will be inflated. This inflation may result in false identification of the population as a putative source for the focal unknown individual. For this reason, you may wish to remove populations that contain individuals likely to be subject to contemporary hybridization or admixture.

Usage

```
gl.assign.on.genotype(
  x,
  unknown,
  nmin = 10,
  n.best = NULL,
  aic.threshold = 0.05,
  verbose = NULL
)
```

Arguments

`x` Name of the input genlight object [required].

`unknown` SpecimenID label (indName) of the focal individual whose provenance is unknown [required].

`nmin` Minimum sample size for a target population to be included in the analysis [default 10].

`n.best` If given a value, dictates the best n=n.best populations to retain for consideration (or more if their are ties) based on AIC weight. If not specified, then the putative source populations identified as possibilities (AIC.wt >= aic.threshold) are retained. [default NULL].

`aic.threshold` The critical value used to select populations for which their is considered some support as a putative source based on AIC weights [default 0.05]

`verbose` Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2 or as specified using gl.set.verbosity].

Value

A genlight object containing the focal individual (assigned to population 'unknown') and putative source populations based on AIC weights If no such populations, the genlight object contains only data for the unknown individual with a warning.

Author(s)

Script: Arthur Georges. Custodian: Arthur Georges – Post to https://groups.google.com/d/forum/dartr

See Also

gl.assign.pca, gl.assign.pa, gl.assign.mahalanobis

Examples

```
## Not run:
# Test run with a focal individual from the Macleay River (EmmacMaclGeor)
# if (isTRUE(getOption("dartR_fbm"))) testset.gl <- gl.gen2fbm(testset.gl)
test <- gl.assign.on.genotype(testset.gl,unknown='UC_00146',nmin=10,verbose=3)

## End(Not run)
```

gl.assign.pa *Use private alleles to identify populations as possible source populations for an individual of unknown provenance.*

Description

This script identifies as putative source populations, those for which the individual has an expected number of private alleles. The putative source populations are retained and returned in a genlight object.

The algorithm calculates an expectation based on the number of private alleles each individual in the putative source population has in comparison with the other members of that population. From the distribution of these values, an expectation is established as a mean and standard deviation. The private alleles possessed by the unknown individual in comparison with the putative source population is compared to this expectation. Significant departures from expectation renders a population unlikely to be the source for the focal unknown individual.

An excessive count of private alleles is an indication that the unknown does not belong to a target population (provided that the sample size is adequate, say >=10).

WARNING: If a putative population is not in Hardy-Weinberg equilibrium, as might occur if it includes F1 hybrids and backcrosses, then the standard deviation for the expectation will be inflated. This inflation may result in false identification of the population as a putative source for the focal unknown individual. For this reason, you may wish to remove populations that contain individuals likely to be subject to contemporary hybridization or admixture.

Usage

```
gl.assign.pa(
  x,
  unknown,
  nmin = 10,
  n.best = NULL,
  alpha = 0.01,
  verbose = NULL
)
```

Arguments

`x`	Name of the input genlight object [required].
`unknown`	SpecimenID label (indName) of the focal individual whose provenance is unknown [required].
`nmin`	Minimum sample size for a target population to be included in the analysis [default 10].
`n.best`	If given a value, dictates the best n=n.best populations to retain for consideration (or more if their are ties) based on private alleles. If not specified, then the putative source populations identified as significant (p < alpha) are retained. [default NULL].

`alpha`	The critical value used to select populations for which the unknown individual has a count of private alleles within expectation [default 0.001]
`verbose`	Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2 or as specified using gl.set.verbosity].

Value

A genlight object containing the focal individual (assigned to population 'unknown') and populations for which the focal individual is not distinctive. If no such populations, the genlight object contains only data for the unknown individual with a warning.

Author(s)

Script: Arthur Georges. Custodian: Arthur Georges – Post to https://groups.google.com/d/forum/dartr

See Also

gl.assign.pca

Examples

```
# Test run with a focal individual from the Macleay River (EmmacMaclGeor)
if (isTRUE(getOption("dartR_fbm"))) testset.gl <- gl.gen2fbm(testset.gl)
#test <- gl.assign.pa(testset.gl,unknown='UC_00146',nmin=10,verbose=3)
```

gl.assign.pca *Eliminate from consideration putative source populations for a specified individual of unknown provenance using PCA*

Description

This script eliminates from consideration putative source populations for a specified individual of unknown provinence based on its proximity to each putative source population defined by a confidence ellipse in ordinated space of two dimensions.

The following process is followed:

1. The space defined by the loci is ordinated to yield a series of orthogonal axes (independent) and the top two dimensions are considered. Populations for which the unknown individual lies outside the specified confidence limits are set aside to allow further examination.

Usage

```
gl.assign.pca(
  x,
  unknown,
  nmin = 10,
  plevel = 0.001,
  plot.out = TRUE,
  verbose = NULL
)
```

Arguments

x	Name of the input genlight object [required].
unknown	Identity label of the focal individual whose provenance is unknown [required].
nmin	Minimum sample size for a target population to be included in the analysis [default 10].
plevel	Probability level for bounding ellipses in the PCoA plot [default 0.999].
plot.out	If TRUE, plot the 2D PCA showing the position of the unknown [default TRUE]
verbose	Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2 or as specified using gl.set.verbosity].

Details

There are three considerations to assignment. First, consider only those populations for which the unknown has no private alleles. Substanial numbers of private alleles are an indication that the unknown does not belong to a target population (provided that the sample size is adequate, say >=10). This can be evaluated with gl.assign.pa().

A next step is to consider the PCA plot for populations where no private alleles have been detected and the position of the unknown in relation to confidence ellipses as produced by this script. Note, this plot is considering only the top two dimensions of the ordination. This is justified because an unknown lying outside the confidence ellipse in two dimensions cannot lie within the confidence envelope incorporating deeper dimensions. It can be unambiguously interpreted as it lying outside the confidence envelope. However, if the unknown lies inside the confidence ellipse in two dimensions, then it may still lie outside the confidence envelope in deeper dimensions.

As with the first step using gl.assign.pa(), this second step is good for eliminating populations from consideration, but does not provide confidence in assignment.

The third step is to consider the assignment probabilities, using the script gl.assign.mahalanobis(). This approach calculates the squared Generalised Linear Distance (Mahalanobis distance) of the unknown from the centroid for each remaining putative source population, and calculates the probability associated with its quantile under the zero truncated normal distribution. This index takes into account position of the unknown in relation to the confidence envelope in all selected dimensions of the ordination.

Each of these approaches provides evidence, none are 100 need to be interpreted cautiously. They are best applied sequentially.

In deciding the assignment, the script considers an individual to be an outlier with respect to a particular population at alpha = 0.001 as default.

Value

A genlight object containing only those populations that are putative source populations for the unknown individual.

Author(s)

Script: Arthur Georges. Custodian: Arthur Georges – Post to https://groups.google.com/d/forum/dartr

---

`gl.diagnostics.relatedness`

Run simulations and relatedness analyses on genlight objects

Description

This function wraps a variety of methods for estimating relatedness, such that they can be directly compared for accuracy and precision. It also provides the ability to run the gl.sim function for a minimum of 3 generations, providing further functionality with regards to estimating gene flow and population dynamics. It supports multiple simulation back ends, correlation output, error checking, RMSE/variance summaries, and optional plotting.

Usage

```
gl.diagnostics.relatedness(
  x,
  cleanup = FALSE,
  ref_variables = NULL,
  sim_variables = NULL,
  which_tests = "wang",
  run_sim = FALSE,
  IncludePlots = FALSE,
  plotOut = FALSE,
  varOut = FALSE,
  rmseOut = FALSE,
  numberIterations = 1,
  numberGenerations = 3,
  genToSave = "all",
  runE9 = FALSE,
  E9Inbreed = FALSE,
  e9Path = NULL,
  verbose = NULL,
  e9parallel = FALSE,
  nCores = 1,
  includedPed = FALSE
)
```

Arguments

`x` A genlight object containing SNP or SilicoDArT data [required].

`cleanup` Logical. Apply callrate, heterozygosity and all-NA filters before simulation [default = FALSE].

`ref_variables` Path to reference variable file [optional].

`sim_variables` Path to simulation variable file [optional].

`which_tests` Character vector of relatedness tests to apply [default = "wang"].

`run_sim` Logical. If TRUE, run simulations [default = FALSE].

`IncludePlots` Logical. If TRUE, generate and return plots [default = FALSE].

`plotOut` Logical. If TRUE, prints plots [default = FALSE].

`varOut` Logical. If TRUE, return variance results [default = FALSE].

`rmseOut` Logical. If TRUE, return RMSE results [default = FALSE].

`numberIterations`
Integer. Number of simulation iterations [default = 1].

`numberGenerations`
Integer. Number of generations to simulate [default = 3].

`genToSave` Either "all" or a numeric vector of generations to save [default = "all"].

`runE9` Logical. If TRUE, include E9 analysis [default = FALSE].

`E9Inbreed` Logical. If TRUE, then runs EMIBD9 twice - once with inbreeding once w/out [default = FALSE].

`e9Path` Path to external E9 binary [optional].

`verbose` Verbosity level: 0–5. If NULL, set by `gl.set.verbosity()` [default = NULL].

`e9parallel` Logical. Run E9 in parallel [default = FALSE].

`nCores` Integer. Number of cores if running E9 in parallel [default = 1].

`includedPed` Logical. If TRUE then input file has attache pedigree [default = FALSE]

Details

The function manages filtering, simulation setup, correlation and relatedness outputs, and optional plotting. It handles quality control checks on input objects and file paths before analysis.

Value

Returns an S4 object containing simulation and/or relatedness outputs. The slots for the output class are as follows:

- @InputDf: Original genlight input
- @SimOutput: Genlight object of simulation outputs
- @corOutList: Results of correlation analysis
- @corVals: Output of correlation results between methods
- @plotList: List of plots

Author(s)

Ethan, Luis (Post to https://groups.google.com/d/forum/dartr)

See Also

gl.filter.callrate, gl.filter.heterozygosity

Examples

```
## Not run:
if (isTRUE(getOption("dartR_fbm"))) testset.gl <- gl.gen2fbm(testset.gl)
gl.diagnostics.relatedness(testset.gl, run_sim = TRUE, IncludePlots = TRUE)

## End(Not run)
```

gl.filter.parent.offspring

Filters putative parent offspring within a population

Description

This script removes individuals suspected of being related as parent-offspring,using the output of the function gl.report.parent.offspring, which examines the frequency of pedigree inconsistent loci, that is, those loci that are homozygotes in the parent for the reference allele, and homozygous in the offspring for the alternate allele. This condition is not consistent with any pedigree, regardless of the (unknown) genotype of the other parent. The pedigree inconsistent loci are counted as an indication of whether or not it is reasonable to propose the two individuals are in a parent-offspring relationship.

Usage

```
gl.filter.parent.offspring(
  x,
  min.rdepth = 12,
  min.reproducibility = 1,
  range = 1.5,
  method = "best",
  rm.monomorphs = FALSE,
  plot_theme = theme_dartR(),
  plot_colors = gl.colors(2),
  plot.file = NULL,
  plot.dir = NULL,
  verbose = NULL
)
```

Arguments

`x` Name of the genlight object containing the SNP genotypes [required].

`min.rdepth` Minimum read depth to include in analysis [default 12].

`min.reproducibility` Minimum reproducibility to include in analysis [default 1].

`range` Specifies the range to extend beyond the interquartile range for delimiting outliers [default 1.5 interquartile ranges].

`method` Method of selecting the individual to retain from each pair of parent offspring relationship, 'best' (based on CallRate) or 'random' [default 'best'].

`rm.monomorphs` If TRUE, remove monomorphic loci after filtering individuals [default FALSE].

`plot_theme` Theme for the plot. See Details for options [default theme_dartR()].

`plot_colors` List of two color names for the borders and fill of the plots [default gl.colors(2)].

`plot.file` Name for the RDS binary file to save (base name only, exclude extension) [default NULL]

`plot.dir` Directory to save the plot RDS files [default as specified by the global working directory or tempdir()]

`verbose` Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log ; 3, progress and results summary; 5, full report [default 2, unless specified using gl.set.verbosity].

Details

If two individuals are in a parent offspring relationship, the true number of pedigree inconsistent loci should be zero, but SNP calling is not infallible. Some loci will be miss-called. The problem thus becomes one of determining if the two focal individuals have a count of pedigree inconsistent loci less than would be expected of typical unrelated individuals. There are some quite sophisticated software packages available to formally apply likelihoods to the decision, but we use a simple outlier comparison.

To reduce the frequency of miss-calls, and so emphasize the difference between true parent-offspring pairs and unrelated pairs, the data can be filtered on read depth. Typically minimum read depth is set to 5x, but you can examine the distribution of read depths with the function `gl.report.rdepth` and push this up with an acceptable loss of loci. 12x might be a good minimum for this particular analysis. It is sensible also to push the minimum reproducibility up to 1, if that does not result in an unacceptable loss of loci. Reproducibility is stored in the slot `@other$loc.metrics$RepAvg` and is defined as the proportion of technical replicate assay pairs for which the marker score is consistent. You can examine the distribution of reproducibility with the function `gl.report.reproducibility`.

Note that the null expectation is not well defined, and the power reduced, if the population from which the putative parent-offspring pairs are drawn contains many sibs. Note also that if an individual has been genotyped twice in the dataset, the replicate pair will be assessed by this script as being in a parent-offspring relationship.

You should run `gl.report.parent.offspring` before filtering. Use this report to decide min.rdepth and min.reproducibility and assess impact on your dataset.

Note that if your dataset does not contain RepAvg or rdepth among the locus metrics, the filters for reproducibility and read depth are no used.

Examples of other themes that can be used can be consulted in

- https://ggplot2.tidyverse.org/reference/ggtheme.html and
- https://yutannihilation.github.io/allYourFigureAreBelongToUs/ggthemes/

Value

the filtered genlight object without A set of individuals in parent-offspring relationship. NULL if no parent-offspring relationships were found.

Author(s)

Custodian: Arthur Georges – Post to https://groups.google.com/d/forum/dartr

See Also

gl.report.rdepth , gl.report.reproducibility, gl.report.parent.offspring

Examples

```
if (isTRUE(getOption("dartR_fbm"))) testset.gl <- gl.gen2fbm(testset.gl)
out <- gl.filter.parent.offspring(testset.gl[1:10, 1:50])
```

gl.grm *Calculates an identity by descent matrix*

Description

This function calculates the mean probability of identity by state (IBS) across loci that would result from all the possible crosses of the individuals analyzed. IBD is calculated by an additive relationship matrix approach developed by Endelman and Jannink (2012) as implemented in the function A.mat (package rrBLUP).

Usage

```
gl.grm(
  x,
  plotheatmap = TRUE,
  palette_discrete = NULL,
  palette_convergent = NULL,
  legendx = 0,
  legendy = 0.5,
  label.size = 0.75,
  legend.title = "Populations",
  plot.file = NULL,
  plot.dir = NULL,
  verbose = NULL,
  ...
)
```

Arguments

`x` Name of the genlight object containing the SNP data [required].

`plotheatmap` A switch if a heatmap should be shown [default TRUE].

`palette_discrete` the color of populations [gl.select.colors].

`palette_convergent` A convergent palette for the IBD values [default convergent_palette].

`legendx` x coordinates for the legend[default 0].

`legendy` y coordinates for the legend[default 1].

`label.size` Specify the size of the population labels [default 0.75].

`legend.title` Legend title [default "Populations"].

`plot.file` Name for the RDS binary file to save (base name only, exclude extension) [default NULL]

`plot.dir` Directory in which to save files [default = working directory]

`verbose` Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log ; 3, progress and results summary; 5, full report [default 2 or as specified using gl.set.verbosity].

`...` Parameters passed to function A.mat from package rrBLUP.

Details

This function uses the A.mat function from the rrBLUP package. This method follows the approach developed by Endelman and Jannink (2012).

Two alleles are Identical by State (IBS) if they are the same in state, regardless of whether they come from a common ancestor. Two alleles are Identical by Descent (IBD) if they are inherited from a common ancestor. While IBS does not necessarily imply IBD, using high-density SNP data improves the estimation of IBD probabilities from IBS measures.

This function also plots a heatmap, and a dendrogram, of IBD values where each diagonal element has a mean that equals 1+f, where f is the inbreeding coefficient (i.e. the probability that the two alleles at a randomly chosen locus are IBD from the base population). As this probability lies between 0 and 1, the diagonal elements range from 1 to 2. Because the inbreeding coefficients are expressed relative to the current population, the mean of the off-diagonal elements is -(1+f)/n, where n is the number of loci. Individual names are shown in the margins of the heatmap and colors represent different populations.

Value

An identity by descent matrix

Author(s)

Custodian: Arthur Georges – Post to https://groups.google.com/d/forum/dartr

References

- Endelman, J. B. (2011). Ridge regression and other kernels for genomic selection with r package rrblup. The Plant Genome 4, 250.
- Endelman, J. B. , Jannink, J.-L. (2012). Shrinkage estimation of the realized relationship matrix. G3: Genes, Genomics, Genetics 2, 1405.

See Also

`gl.grm.network`

Other inbreeding functions: `gl.grm.network()`

Examples

```
if (isTRUE(getOption("dartR_fbm"))) platypus.gl <- gl.gen2fbm(platypus.gl)
gl.grm(platypus.gl[1:10, 1:100])
```

---

gl.grm.network *Represents a similarity matrix as a network*

Description

This script takes any similarity matrix and represents the relationship among the specimens as a network diagram.

Usage

```
gl.grm.network(
  G,
  x,
  standardise = FALSE,
  categorise = FALSE,
  color.categories = c("#E63E94", "#E5D44C", "#3ED2E6"),
  method = "fr",
  node.size = 8,
  node.label = TRUE,
  node.label.size = 2,
  node.label.color = "black",
  link.color = NULL,
  link.size = 2,
  kinship.threshold = 0.125,
  title = "Network of a similarity matrix",
  legend.title = "Populations",
  title.size = 16,
  legend.size = 14,
  palette_discrete = NULL,
  plot.dir = NULL,
  plot.file = NULL,
  verbose = NULL
)
```

Arguments

- `G` A similarity matrix [required].
- `x` A genlight object from which the matrix was generated [required].
- `standardise` Whether to standardise matrix using Goudet et al method, see details [default FALSE].
- `categorise` Whether to categorise the color of the link representing kinship values into relationships. Same Individual (>0.3), Full Siblings / Parent-Offspring (>0.2 & <0.3) and Half Siblings (>0.1 & <0.2) [default FALSE].
- `color.categories` A vector of three colors to represent the above kinship categories [default = c("#E63E94","#E5D44C","#3ED2E6")].
- `method` One of 'fr', 'kk', 'gh' or 'mds' [default 'fr'].
- `node.size` Size of the symbols for the network nodes [default 8].
- `node.label` TRUE to display node labels [default TRUE].
- `node.label.size` Size of the node labels [default 3].
- `node.label.color` Color of the text of the node labels [default 'black'].
- `link.color` Colors for links, either a vector of colors or a color palette function [NULL].
- `link.size` Size of the links [default 2].
- `kinship.threshold` Threshold of kinship value to display in the network diagram [default 0.125].
- `title` Title for the plot [default 'Network of similarity matrix'].
- `legend.title` Title for the legend [default "Populations"].
- `title.size` Font size of the title [default 16].
- `legend.size` Font size of the legend [default 14].
- `palette_discrete` A discrete set of colors with as many colors as there are populations in the dataset [default NULL].
- `plot.dir` Directory to save the plot RDS files [default as specified by the global working directory or tempdir]
- `plot.file` Name for the RDS binary file to save (base name only, exclude extension) [default NULL]
- `verbose` Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log ; 3, progress and results summary; 5, full report [default 2 or as specified using gl.set.verbosity].

Details

The gl.grm.network function creates a network diagram that represents genetic relationships among individuals in a dataset.

Layout options

Four layout options are implemented in this function:

- 'fr' Fruchterman-Reingold layout layout_with_fr (package igraph)
- 'kk' Kamada-Kawai layout layout_with_kk (package igraph)
- 'gh' Graphopt layout layout_with_graphopt (package igraph)
- 'mds' Multidimensional scaling layout layout_with_mds (package igraph)

Standardise matrix using Goudet et al method

Choosing meaningful thresholds to represent relationships between individuals can be challenging because kinship and inbreeding coefficients are relative measures. To standardize a genomic relationship matrix (GRM), such as the one produced by the function gl.grm, and facilitate interpretation, the function adjusts the matrix through the following steps:

1. Centering Inbreeding Coefficients: Subtract 1 from the mean of the diagonal elements to calculate the average inbreeding coefficient. This centers the inbreeding coefficients around zero, providing a reference point relative to the population's average inbreeding level.

2. Calculating Kinship Coefficients: Divide the off-diagonal elements by 2 to obtain the kinship coefficients. This conversion reflects the probability of sharing alleles IBD between pairs of individuals.

3. Centering Kinship Coefficients: Subtract the adjusted mean inbreeding coefficient (from step 1) from each kinship coefficient (from step 2). This centers the kinship coefficients relative to the population average, allowing for meaningful comparisons.

This adjustment method aligns with the approach used by Goudet et al. (2018), enabling the relationships to be interpreted in the context of the overall genetic relatedness within the population.

Below is a table modified from Speed & Balding (2015) showing kinship values, and their confidence intervals (CI), for different relationships that could be used to guide the choosing of the kinship threshold in the function.

Relationship	**Kinship**	**95% CI**
Identical twins / clones / same individual	0.5	–
Sibling / Parent–Offspring	0.25	(0.204, 0.296)
Half-sibling	0.125	(0.092, 0.158)
First cousin	0.062	(0.038, 0.089)
Half-cousin	0.031	(0.012, 0.055)
Second cousin	0.016	(0.004, 0.031)
Half-second cousin	0.008	(0.001, 0.020)
Third cousin	0.004	(0.000, 0.012)
Unrelated	0	–

Value

A network plot showing kinship between individuals

Author(s)

Custodian: Arthur Georges – Post to https://groups.google.com/d/forum/dartr

References

- Endelman, J. B. , Jannink, J.-L. (2012). Shrinkage estimation of the realized relationship matrix. G3: Genes, Genomics, Genetics 2, 1405.
- Goudet, J., Kay, T., & Weir, B. S. (2018). How to estimate kinship. Molecular Ecology, 27(20), 4121-4135.
- Speed, D., & Balding, D. J. (2015). Relatedness in the post-genomic era: is it still useful?. Nature Reviews Genetics, 16(1), 33-44.

See Also

gl.grm

Other inbreeding functions: gl.grm()

Examples

```
if (requireNamespace("igraph", quietly = TRUE) & requireNamespace("rrBLUP",
  quietly = TRUE
) & requireNamespace("fields", quietly = TRUE)) {
  if (isTRUE(getOption("dartR_fbm"))) possums.gl <- gl.gen2fbm(possums.gl)
  t1 <- possums.gl
  # filtering on call rate
  t1 <- gl.filter.callrate(t1)
  t1 <- gl.subsample.loc(t1, n = 100)
  # relatedness matrix
  res <- gl.grm(t1, plotheatmap = FALSE)
  # relatedness network
  res2 <- gl.grm.network(res, t1, kinship.threshold = 0.125)
}
```

---

gl.plot.network *Represents a distance or dissimilarity matrix as a network*

Description

This script takes a distance matrix generated by dist() and represents the relationship among the specimens as a network diagram. In order to use this script, a decision is required on a threshold for relatedness to be represented as link in the network, and on the layout used to create the diagram.

Usage

```
gl.plot.network(
  D,
  x = NULL,
  method = "fr",
  node.size = 3,
  node.label = FALSE,
  node.label.size = 0.7,
  node.label.color = "black",
  alpha = 0.005,
  title = "Network based on genetic distance",
  verbose = NULL
)
```

Arguments

`D`	A distance or dissimilarity matrix generated by dist() or gl.dist() [required].
`x`	A genlight object from which the D matrix was generated [default NULL].
`method`	One of "fr", "kk" or "drl" [default "fr"].
`node.size`	Size of the symbols for the network nodes [default 3].
`node.label`	TRUE to display node labels [default FALSE].
`node.label.size`	Size of the node labels [default 0.7].
`node.label.color`	Color of the text of the node labels [default 'black'].
`alpha`	Upper threshold to determine which links between nodes to display [default 0.005].
`title`	Title for the plot [default "Network based on genetic distance"].
`verbose`	Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2, unless specified using gl.set.verbosity].

Details

The threshold for relatedness to be represented as a link in the network is specified as a quantile. Those relatedness measures above the quantile are plotted as links, those below the quantile are not. Often you are looking for relatedness outliers in comparison with the overall relatedness among individuals, so a very conservative quantile is used (e.g. 0.004), but ultimately, this decision is made as a matter of trial and error. One way to approach this trial and error is to try to achieve a sparse set of links between unrelated 'background' individuals so that the stronger links are preferentially shown.

There are several layouts from which to choose. The most popular are given as options in this script.

- fr – Fruchterman, T.M.J. and Reingold, E.M. (1991). Graph Drawing by Force-directed Placement. Software – Practice and Experience 21:1129-1164.
- kk – Kamada, T. and Kawai, S.: An Algorithm for Drawing General Undirected Graphs. Information Processing Letters 31:7-15, 1989.

- drl – Martin, S., Brown, W.M., Klavans, R., Boyack, K.W., DrL: Distributed Recursive (Graph) Layout. SAND Reports 2936:1-10, 2008.

Colors of node symbols are those of the rainbow.

Value

returns no value (i.e. NULL)

Author(s)

Custodian: Arthur Georges – Post to https://groups.google.com/d/forum/dartr

Examples

```
if ((requireNamespace("rrBLUP", quietly = TRUE)) & (requireNamespace("gplots", quietly = TRUE))) {
  if (isTRUE(getOption("dartR_fbm"))) platypus.gl <- gl.gen2fbm(platypus.gl)
  test <- gl.subsample.loc(platypus.gl, n = 100)
  test <- gl.keep.ind(test, ind.list = indNames(test)[1:10])
  D <- gl.grm(test, legendx = 0.04)
  gl.plot.network(D, test)
}
```

gl.report.parent.offspring

Identifies putative parent offspring within a population

Description

This script examines the frequency of pedigree inconsistent loci, that is, those loci that are homozygotes in the parent for the reference allele, and homozygous in the offspring for the alternate allele. This condition is not consistent with any pedigree, regardless of the (unknown) genotype of the other parent. The pedigree inconsistent loci are counted as an indication of whether or not it is reasonable to propose the two individuals are in a parent-offspring relationship.

Usage

```
gl.report.parent.offspring(
  x,
  min.rdepth = 12,
  min.reproducibility = 1,
  range = 1.5,
  plot.filters = FALSE,
  plot_theme = theme_dartR(),
  plot_colors = gl.colors(2),
  plot.dir = NULL,
  plot.file = NULL,
  verbose = NULL
)
```

Arguments

`x` Name of the genlight object containing the SNP genotypes [required].

`min.rdepth` Minimum read depth to include in analysis [default 12].

`min.reproducibility` Minimum reproducibility to include in analysis [default 1].

`range` Specifies the range to extend beyond the interquartile range for delimiting outliers [default 1.5 interquartile ranges].

`plot.filters` Whether to show the plots of filters within the function [default FALSE].

`plot_theme` Theme for the plot. See Details for options [default theme_dartR()].

`plot_colors` List of two color names for the borders and fill of the plots [default gl.colors(2)].

`plot.dir` Directory to save the plot RDS files [default as specified by the global working directory or tempdir()]

`plot.file` Name for the RDS binary file to save (base name only, exclude extension) [default NULL] Creates a plot that shows the sex linked markers.

`verbose` Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2, unless specified using gl.set.verbosity].

Details

If two individuals are in a parent offspring relationship, the true number of pedigree inconsistent loci should be zero, but SNP calling is not infallible. Some loci will be miss-called. The problem thus becomes one of determining if the two focal individuals have a count of pedigree inconsistent loci less than would be expected of typical unrelated individuals. There are some quite sophisticated software packages available to formally apply likelihoods to the decision, but we use a simple outlier comparison.

To reduce the frequency of miss-calls, and so emphasize the difference between true parent-offspring pairs and unrelated pairs, the data can be filtered on read depth.

Typically minimum read depth is set to 5x, but you can examine the distribution of read depths with the function `gl.report.rdepth` and push this up with an acceptable loss of loci. 12x might be a good minimum for this particular analysis. It is sensible also to push the minimum reproducibility up to 1, if that does not result in an unacceptable loss of loci. Reproducibility is stored in the slot `@other$loc.metrics$RepAvg` and is defined as the proportion of technical replicate assay pairs for which the marker score is consistent. You can examine the distribution of reproducibility with the function `gl.report.reproducibility`.

Note that the null expectation is not well defined, and the power reduced, if the population from which the putative parent-offspring pairs are drawn contains many sibs. Note also that if an individual has been genotyped twice in the dataset, the replicate pair will be assessed by this script as being in a parent-offspring relationship.

The function `gl.filter.parent.offspring` will filter out those individuals in a parent offspring relationship.

Note that if your dataset does not contain RepAvg or rdepth among the locus metrics, the filters for reproducibility and read depth are no used. Examples of other themes that can be used can be consulted in

- `https://ggplot2.tidyverse.org/reference/ggtheme.html` and
- `https://yutannihilation.github.io/allYourFigureAreBelongToUs/ggthemes/`

Value

A set of individuals in parent-offspring relationship. NULL if no parent-offspring relationships were found.

Author(s)

Custodian: Arthur Georges (Post to https://groups.google.com/d/forum/dartr)

See Also

gl.report.rdepth ,gl.report.reproducibility, gl.filter.parent.offspring

Examples

```
if (isTRUE(getOption("dartR_fbm"))) testset.gl <- gl.gen2fbm(testset.gl)
out <- gl.report.parent.offspring(testset.gl[1:10, 1:100])
```

---

gl.run.colony *Run COLONY2*

Description

A convenient R wrapper for the COLONY pedigree-inference software (Jones & Wang 2010), allowing users to perform full-pedigree likelihood analyses of multilocus genotype data directly from R. This function automates the creation of the required ‘Colony2.DAT‘ input file and runs the COLONY executable.

Usage

```
gl.run.colony(
  x,
  colony.path = getwd(),
  outfile = "colony2.dat",
  outpath = NULL,
  project.name = "my_project",
  output.name = "my_project",
  probability.father = 0.5,
  probability.mother = 0.5,
  seed = NULL,
  update.allele.freq = 0,
  di.mono.ecious = 2,
  inbreed = 0,
  haplodiploid = 0,
  polygamy.male = 0,
  polygamy.female = 0,
  clone.inference = 1,
  scale.shibship = 1,
  sibship.prior = 0,
  known.allele.freq = 0,
  num.runs = 1,
  length.run = 2,
  monitor.method = 0,
  monitor.interval = 10000,
  windows.gui = 0,
  likelihood = 0,
  precision.fl = 2,
  marker.id = "mk@",
  marker.type = "0@",
  allelic.dropout = "0.000@",
  other.typ.err = "0.05@",
  paternity.exclusion.threshold = "0 0",
  maternity.exclusion.threshold = "0 0",
  paternal.sibship = 0,
  maternal.sibship = 0,
  excluded.paternity = 0,
  excluded.maternity = 0,
  excluded.paternal.sibships = 0,
  excluded.maternity.sibships = 0,
  verbose = NULL
)
```

Arguments

`x` A `genlight` object with individual metadata columns 'offspring', 'mother', and 'father' indicating 'yes'/'no' for each sample [required].

`colony.path` Path to the colony executable [default getwd()].

`outfile` File name of the output file (including extension) [default "colony2.dat"].

`outpath` Path where to save the output file [default global working directory or if not specified, tempdir()].

`project.name` Project name to include in the file header [default 'my_project'].

`output.name` Output name to include in the file header [default 'my_project'].

`probability.father` Probability that the father of an offspring is included among candidates [default 0.5].

`probability.mother` Probability that the mother of an offspring is included among candidates [default 0.5].

`seed` Seed for the random number generator [default NULL].

`update.allele.freq` 0 = do not update allele frequencies; 1 = update [default 0].

`di.mono.ecious` 2 = dioecious species; 1 = monoecious species [default 2].

`inbreed` 0 = no inbreeding; 1 = inbreeding allowed [default 0].

`haplodiploid` 0 = diploid species; 1 = haplodiploid species [default 0].

`polygamy.male` 0 = polygamy; 1 = monogamy for males [default 0].

`polygamy.female` 0 = polygamy; 1 = monogamy for females [default 0].

`clone.inference` 0 = no clone inference; 1 = infer clones [default 1].

`scale.shibship` 0 = do not scale full sibship; 1 = scale [default 1].

`sibship.prior` 0–4 specifying sibship prior strength (No, Weak, Medium, Strong, Optimal) [default 0].

`known.allele.freq` 0 = unknown allele frequencies; 1 = known [default 0].

`num.runs` Number of runs [default 1].

`length.run` 1–4 specifying run length (short, medium, long, very long) [default 2].

`monitor.method` 0 = monitor by iteration number; 1 = monitor by time (seconds) [default 0].

`monitor.interval` Interval for monitoring (either iteration count or seconds) [default 10000].

`windows.gui` 0 = no Windows GUI; 1 = use Windows GUI [default 0].

`likelihood` 0–2 specifying likelihood scoring (PairLikelihood, FullLikelihood, FPLS) [default 0].

`precision.fl` 0–3 specifying precision level for full-likelihood (Low, Medium, High, Very-High) [default 2].

`marker.id` Marker IDs string for all loci [default ’mk@’].

`marker.type` Marker types string for all loci (0@ for codominant, 1@ for dominant) [default ’0@’].

`allelic.dropout` Allelic dropout rate string per locus [default ’0.000@’].

`other.typ.err` Other typing error rate string per locus [default ’0.05@’].

`paternity.exclusion.threshold` Threshold for paternity exclusion ("0 0") [default ’0 0’].

`maternity.exclusion.threshold` Threshold for maternity exclusion ("0 0") [default ’0 0’].

`paternal.sibship` Number of known paternal sibships [default 0].

`maternal.sibship` Number of known maternal sibships [default 0].

`excluded.paternity` Number of offspring with excluded paternity [default 0].

`excluded.maternity` Number of offspring with excluded maternity [default 0].

`excluded.paternal.sibships` Number of excluded paternal sibships [default 0].

`excluded.maternity.sibships` Number of excluded maternal sibships [default 0].

`verbose` Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log ; 3, progress and results summary; 5, full report [default 2 or as specified using gl.set.verbosity].

Details

COLONY implements a Bayesian full-pedigree likelihood method that simultaneously infers sibships and parentage by considering the likelihood of entire pedigree configurations rather than pairwise comparisons.

Value

Invisibly returns the output filename.

Author(s)

Jesús Castrejón-Figueroa, Diana A. Robledo-Ruiz & Luis Mijangos– Post to https://groups.google.com/d/forum/dartr

References

Wang, J. (2011). COLONY: a program for parentage and sibship inference from multilocus genotype data. Molecular Ecology Resources 10: 551–555.

Examples

```
## Not run:
if (isTRUE(getOption("dartR_fbm"))) testset.gl <- gl.gen2fbm(testset.gl)
gl2colony(x = testset.gl)

## End(Not run)
```

---

gl.run.EMIBD9 *Run program EMIBD9*

Description

Run program EMIBD9

Usage

```
gl.run.EMIBD9(
  x,
  outfile = "EMIBD9_Res.ibd9",
  outpath = tempdir(),
  emibd9.path = getwd(),
  OutAlleleFre = 0,
  EM_Method = 1,
  Inbreed = FALSE,
  palette_convergent = NULL,
  parallel = FALSE,
  ncores = 1,
  ISeed = 42,
  plot.out = TRUE,
  plot.dir = NULL,
  plot.file = NULL,
  verbose = NULL
)
```

Arguments

`x` Name of the genlight object containing the SNP data [required].

`outfile` A string, giving the path and name of the output file [default "EMIBD9_Res.ibd9"].

`outpath` Path where to save the output file. Use outpath=getwd() or outpath='.' when calling this function to direct output files to your working or current directory [default tempdir(), mandated by CRAN].

`emibd9.path` Path to the folder emidb files. Please note there are 2 different executables depending on your OS: EM_IBD_P.exe (=Windows) EM_IBD_P (=Mac, Linux). You only need to point to the folder (the function will recognise which OS you are running) [default getwd()].

`OutAlleleFre` A boolean that indicates whether to output allele frequencies [default FALSE].

`EM_Method` An integer that indicates the method to use for the expectation maximization (EM) algorithm. 1, the standard EM method; 2, the EM method with a quasi-Newton acceleration; 3, the EM method with a SQUAREM acceleration [default 1].

`Inbreed` A boolean that indicates whether to compute inbreeding (i.e. delta1 to delta6) [default FALSE].

`palette_convergent` A character vector of colours to use for the heatmap plot. If NULL, the default palette from gl.colors("div") will be used [default NULL].

`parallel` A boolean that indicates whether to run the parallel version of EM IBD9 (EM_IBD_P_mpi) [default FALSE].

`ncores` An integer specifying the number of cores to use when parallel is TRUE [default 1].

`ISeed` An integer specifying the random seed to use for the EM algorithm [default 42].

`plot.out` A boolean that indicates whether to plot the results [default TRUE].

`plot.dir` Directory to save the plot RDS files [default as specified by the global working directory or tempdir()]

`plot.file` Name for the RDS binary file to save (base name only, exclude extension) [default NULL]

`verbose` Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default NULL, unless specified using gl.set.verbosity]

Details

The results of EMIBD9 include the identical in state (IIS) values for each mode (S1 - 9) and nine condensed identical by descent (IBD) modes (delta1 - delta9) as well as the relatedness coefficient (r). Alleles are IIS if they are the same. Similarly, IBD describes a matching allele between two individuals that has been inherited from a common ancestor or common gene. In a pairwise comparison, delta1 to delta9 are the probabilities associated with each IBD mode. delta1 to delta6 take vakue > 0 in presence of inbreeding and hence are only computed when this option is selected.

EMIBD9 uses an expectation maximization (EM) algorithm based on the maximum likelihood expectations (MLE) of δ to estimate both allele frequencies (p) and δ jointly from genotype data. By iteratively calculating p and δ, relatedness can be modified to reduce biases due to small sample sizes. Wang J. (2022) suggest the resulting r coefficient is therefore more robust compared to previous methods.

The kinship coefficient is the probability that two alleles at a random locus drawn from two individuals are IBD.

Below is a table modified from Speed & Balding (2015) showing kinship values, and their confidence intervals (CI), for different relationships.

Relationship	**Kinship**	**95% CI**
Identical twins / clones / same individual	0.5	–
Sibling / Parent–Offspring	0.25	(0.204, 0.296)
Half-sibling	0.125	(0.092, 0.158)
First cousin	0.062	(0.038, 0.089)
Half-cousin	0.031	(0.012, 0.055)
Second cousin	0.016	(0.004, 0.031)
Half-second cousin	0.008	(0.001, 0.020)
Third cousin	0.004	(0.000, 0.012)
Unrelated	0	–

For greater detail on the methods employed by EMIBD9, we encourage you to read Wang, J. (2022).

Download the program from here:

https://www.zsl.org/about-zsl/resources/software/emibd9

For Windows, Mac and Linux install the program then point to the folder where you find: EM_IBD_P.exe (=Windows) and EM_IBD_P (=Mac, Linux). If running really slow you may want to create the files using the function and then run in parallel using the documentation provided by the authors [you need to have mpiexec installed].

Please note individual names must have a maximal length of 20 characters. The IDs must NOT contain blank space and other illegal characters (such as /), and must be unique among all sampled individuals (i.e. NO duplications). Any string longer than 20 characters for individual ID will be truncated to have 20 characters.

Value

A list with three or four elements depending on whether inbreeding was selected. The first element (rel) is a matrix with pairwise relatedness. The second (raw) is the raw output table from the program. The third (processed) is the 'processed' output from the table (self-comparisons - an individuals with itself - and redundant pairs - e.g. the second individuals with the first, when the first vs the second is already present in the results - are removed). The last (inbreeding) is a table of individual inbreeding values (if requested).

Author(s)

Custodian: Luis Mijangos – Post to https://groups.google.com/d/forum/dartr

References

- Wang, J. (2022). A joint likelihood estimator of relatedness and allele frequencies from a small sample of individuals. Methods in Ecology and Evolution, 13(11), 2443-2462.

Examples

```
## Not run:
#To run this function needs EMIBD9 installed in your computer
if (isTRUE(getOption("dartR_fbm"))) platypus.gl <- gl.gen2fbm(platypus.gl)
t1 <- gl.filter.allna(platypus.gl)
res_rel <- gl.run.EMIBD9(t1)

## End(Not run)
```

`gl.sim.relatedness` *Simulate relatedness estimates.*

Description

A simulation based tool to estimate different degrees of relatedness using genlight object to bootstrap the results of kinship estimates. This method uses EMIBD9 (Wang, J. 2022).

Below is a table modified from Speed & Balding (2015) showing kinship values, and their confidence intervals (CI), for different relationships that could be used to guide the choosing of the relatedness threshold in the function.

Relationship	**Kinship**	**95% CI**
Identical twins / clones / same individual	0.5	–
Sibling / Parent–Offspring	0.25	(0.204, 0.296)
Half-sibling	0.125	(0.092, 0.158)
First cousin	0.062	(0.038, 0.089)
Half-cousin	0.031	(0.012, 0.055)
Second cousin	0.016	(0.004, 0.031)
Half-second cousin	0.008	(0.001, 0.020)
Third cousin	0.004	(0.000, 0.012)
Unrelated	0	–

Usage

```
gl.sim.relatedness(
  x,
  rel = "full.sib",
  nboots = 10,
  emibd9.path = getwd(),
  conf = 0.95,
  OutAlleleFre = 0,
  EM_Method = 1,
  Inbreed = FALSE,
  ISeed = 42,
  parallel = FALSE,
  ncores = 1,
  plot.out = TRUE,
  plot.dir = NULL,
  plot.file = NULL,
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object containing the SNP data [required].
`rel`	The degree of relatedness you wish to simulate. One of, 'full.sib', 'half.sib','first.cousin' [default 'full.sib'].
`nboots`	The number of simulation replicates you wish to perform [default 10].
`emibd9.path`	The location of all necessary files to run EMIBD9 (read more at gl.run.EMIBD9) [required].
`conf`	The specified threshold for confidence interval calculation from simulated relatedness values [default 0.95].
`OutAlleleFre`	Whether to write , 1, or not, 0, the allele frequency file [default 0].
`EM_Method`	Whether to estimate delta only (EM_Method=0) or to estimate delta and p jointly (EM_Method=1) [default 1].
`Inbreed`	A Boolean, taking values TRUE or FALSE to indicate inbreeding is not and is allowed in estimating IBD coefficients [default FALSE].
`ISeed`	An integer used to seed the random number generator [default 42].
`parallel`	Use parallelisation. Only works for Mac and Linux at the moment[default FALSE].
`ncores`	How many cores should be used [default 1].
`plot.out`	A boolean that indicates whether to plot the results [default TRUE].
`plot.dir`	Directory to save the plot RDS files [default as specified by the global working directory or tempdir()]
`plot.file`	Name for the RDS binary file to save (base name only, exclude extension) [default NULL]
`verbose`	Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default NULL, unless specified using gl.set.verbosity]

Value

Summary statistics of chosen relatedness relationship and a histogram of relatedness values showing the mean.

Author(s)

Custodian: Sam Amini – Post to https://groups.google.com/d/forum/dartr

References

- Wang, J. (2022). A joint likelihood estimator of relatedness and allele frequencies from a small sample of individuals. Methods in Ecology and Evolution, 13(11), 2443-2462.

  Speed, D., Balding, D. Relatedness in the post-genomic era: is it still useful?. Nat Rev Genet 16, 33–44 (2015).

Examples

```
## Not run:
#To run this function needs EMIBD9 installed in your computer
if (isTRUE(getOption("dartR_fbm"))) platypus.gl <- gl.gen2fbm(platypus.gl)
gl.sim.relatedness(platypus.gl)

## End(Not run)
```

gl2colony *Export a COLONY2 input file from a genlight object*

Description

Export a formatted text file compatible with the COLONY2 software from a genlight object containing parental and offspring information stored in the individual metadata.

Usage

```
gl2colony(
  x,
  outfile = "colony2.dat",
  outpath = NULL,
  project.name = "my_project",
  output.name = "my_project",
  probability.father = 0.5,
  probability.mother = 0.5,
  seed = NULL,
  update.allele.freq = 0,
  di.mono.ecious = 2,
  inbreed = 0,
  haplodiploid = 0,
  polygamy.male = 0,
  polygamy.female = 0,
  clone.inference = 1,
  scale.shibship = 1,
  sibship.prior = 0,
  known.allele.freq = 0,
  num.runs = 1,
  length.run = 2,
  monitor.method = 0,
  monitor.interval = 10000,
  windows.gui = 0,
  likelihood = 0,
  precision.fl = 2,
  marker.id = "mk@",
  marker.type = "0@",
  allelic.dropout = "0.000@",
  other.typ.err = "0.05@",
  paternity.exclusion.threshold = "0 0",
  maternity.exclusion.threshold = "0 0",
  paternal.sibship = 0,
  maternal.sibship = 0,
  excluded.paternity = 0,
  excluded.maternity = 0,
  excluded.paternal.sibships = 0,
  excluded.maternity.sibships = 0,
  verbose = NULL
)
```

Arguments

`x` A `genlight` object with individual metadata columns ’offspring’, ’mother’, and ’father’ indicating ’yes’/’no’ for each sample [required].

`outfile` File name of the output file (including extension) [default "colony2.dat"].

`outpath` Path where to save the output file [default global working directory or if not specified, tempdir()].

`project.name` Project name to include in the file header [default ’my_project’].

`output.name` Output name to include in the file header [default ’my_project’].

`probability.father` Probability that the father of an offspring is included among candidates [default 0.5].

`probability.mother` Probability that the mother of an offspring is included among candidates [default 0.5].

`seed` Seed for the random number generator [default NULL].

`update.allele.freq` 0 = do not update allele frequencies; 1 = update [default 0].

`di.mono.ecious` 2 = dioecious species; 1 = monoecious species [default 2].

`inbreed` 0 = no inbreeding; 1 = inbreeding allowed [default 0].

`haplodiploid` 0 = diploid species; 1 = haplodiploid species [default 0].

`polygamy.male` 0 = polygamy; 1 = monogamy for males [default 0].

`polygamy.female` 0 = polygamy; 1 = monogamy for females [default 0].

`clone.inference` 0 = no clone inference; 1 = infer clones [default 1].

`scale.shibship` 0 = do not scale full sibship; 1 = scale [default 1].

`sibship.prior` 0–4 specifying sibship prior strength (No, Weak, Medium, Strong, Optimal) [default 0].

`known.allele.freq` 0 = unknown allele frequencies; 1 = known [default 0].

`num.runs` Number of runs [default 1].

`length.run` 1–4 specifying run length (short, medium, long, very long) [default 2].

`monitor.method` 0 = monitor by iteration number; 1 = monitor by time (seconds) [default 0].

`monitor.interval` Interval for monitoring (either iteration count or seconds) [default 10000].

`windows.gui` 0 = no Windows GUI; 1 = use Windows GUI [default 0].

`likelihood` 0–2 specifying likelihood scoring (PairLikelihood, FullLikelihood, FPLS) [default 0].

`precision.fl` 0–3 specifying precision level for full-likelihood (Low, Medium, High, Very-High) [default 2].

`marker.id` Marker IDs string for all loci [default 'mk@'].

`marker.type` Marker types string for all loci (0@ for codominant, 1@ for dominant) [default '0@'].

`allelic.dropout` Allelic dropout rate string per locus [default '0.000@'].

`other.typ.err` Other typing error rate string per locus [default '0.05@'].

`paternity.exclusion.threshold` Threshold for paternity exclusion ("0 0") [default '0 0'].

`maternity.exclusion.threshold` Threshold for maternity exclusion ("0 0") [default '0 0'].

`paternal.sibship` Number of known paternal sibships [default 0].

`maternal.sibship` Number of known maternal sibships [default 0].

`excluded.paternity` Number of offspring with excluded paternity [default 0].

`excluded.maternity` Number of offspring with excluded maternity [default 0].

`excluded.paternal.sibships` Number of excluded paternal sibships [default 0].

`excluded.maternity.sibships` Number of excluded maternal sibships [default 0].

`verbose` Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log ; 3, progress and results summary; 5, full report [default 2 or as specified using gl.set.verbosity].

Details

This function formats and writes a COLONY2-compatible text file, including header, offspring genotypes, parental candidate probabilities, and candidate genotypes, based on the genlight object's individual metadata and genotype matrix.

Value

Invisibly returns the output filename.

Author(s)

Jesús Castrejón-Figueroa, Diana A. Robledo-Ruiz – Post to https://groups.google.com/d/forum/dartr

References

Wang, J. (2011). COLONY: a program for parentage and sibship inference from multilocus genotype data. Molecular Ecology Resources 10: 551–555.

Examples

```
## Not run:
if (isTRUE(getOption("dartR_fbm"))) platypus.gl <- gl.gen2fbm(platypus.gl)
gl2colony(x = platypus.gl,
            project.name = "parentage_fish_2022",
            output.name = "parentage_fish_jul_2022",
            seed = 1234,
            probability.father = 0.6,
            probability.mother = 0.4,
            update.allele.freq = 1,
            allelic.dropout = '0.01',
            other.typ.err = '0.001')

## End(Not run)
```

utils.assignment *Population assignment probabilities*

Description

This function takes one individual and estimates their probability of coming from individual populations from multilocus genotype frequencies.

Usage

```
utils.assignment(x, unknown, verbose = NULL)
```

Arguments

`x`	Name of the genlight object containing the SNP data [required].
`unknown`	Name of the individual to be assigned to a population [required].
`verbose`	Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2, unless specified using gl.set.verbosity].

Details

This function is a re-implementation of the function multilocus_assignment from package gstudio. Description of the method used in this function can be found at: https://dyerlab.github.io/applied_population_genetics/populat
assignment.html

Value

A `data.frame` consisting of assignment probabilities for each population.

Author(s)

Custodian: Luis Mijangos – Post to `https://groups.google.com/d/forum/dartr`

Examples

```
require("dartR.data")
if (isTRUE(getOption("dartR_fbm"))) platypus.gl <- gl.gen2fbm(platypus.gl)
res <- utils.assignment(platypus.gl, unknown = "T27")
```

`utils.assignment_2` *Population assignment probabilities*

Description

This function takes one individual and estimates their probability of coming from individual populations from multilocus genotype frequencies.

Usage

```
utils.assignment_2(x, unknown, verbose = NULL)
```

Arguments

`x`	Name of the genlight object containing the SNP data [required].
`unknown`	Name of the individual to be assigned to a population [required].
`verbose`	Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2, unless specified using gl.set.verbosity].

Details

This function is a re-implementation of the function multilocus_assignment from package gstudio. Description of the method used in this function can be found at: https://dyerlab.github.io/applied_population_genetics/populat
assignment.html

Value

A `data.frame` consisting of assignment probabilities for each population.

Author(s)

Custodian: Luis Mijangos – Post to `https://groups.google.com/d/forum/dartr`

Examples

```
require("dartR.data")
if (isTRUE(getOption("dartR_fbm"))) platypus.gl <- gl.gen2fbm(platypus.gl)
res <- utils.assignment_2(platypus.gl, unknown = "T27")
```

`utils.assignment_3` *Population assignment probabilities*

Description

This function takes one individual and estimates their probability of coming from individual populations from multilocus genotype frequencies.

Usage

```
utils.assignment_3(x, unknown, verbose = 2)
```

Arguments

`x`	Name of the genlight object containing the SNP data [required].
`unknown`	Name of the individual to be assigned to a population [required].
`verbose`	Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2, unless specified using gl.set.verbosity].

Details

This function is a re-implementation of the function multilocus_assignment from package gstudio. Description of the method used in this function can be found at: https://dyerlab.github.io/applied_population_genetics/populat
assignment.html

Value

A `data.frame` consisting of assignment probabilities for each population.

Author(s)

Custodian: Luis Mijangos – Post to https://groups.google.com/d/forum/dartr

Examples

```
require("dartR.data")
if (isTRUE(getOption("dartR_fbm"))) platypus.gl <- gl.gen2fbm(platypus.gl)
res <- utils.assignment_2(platypus.gl, unknown = "T27")
```

`utils.assignment_4` *Population assignment probabilities*

Description

This function takes one individual and estimates their probability of coming from individual populations from multilocus genotype frequencies.

Usage

```
utils.assignment_4(x, unknown, verbose = 2)
```

Arguments

`x` Name of the genlight object containing the SNP data [required].

`unknown` Name of the individual to be assigned to a population [required].

`verbose` Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2, unless specified using gl.set.verbosity].

Details

This function is a re-implementation of the function multilocus_assignment from package gstudio. Description of the method used in this function can be found at: https://dyerlab.github.io/applied_population_genetics/populat
assignment.html

Value

A `data.frame` consisting of assignment probabilities for each population.

Author(s)

Custodian: Luis Mijangos – Post to https://groups.google.com/d/forum/dartr

Examples

```
require("dartR.data")
if (isTRUE(getOption("dartR_fbm"))) platypus.gl <- gl.gen2fbm(platypus.gl)
res <- utils.assignment_2(platypus.gl, unknown = "T27")
```

Index