Package 'dartR.captive'

February 18, 2025

Type Package

Title Analysing 'SNP' Data to Support Captive Breeding

Version 1.0.2

Date 2025-02-17

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), adegenet (>= 2.0.0), dartR.base, dartR.data,
dartR.sim

Imports methods, utils, crayon, ggplot2, patchwork, stringr,
data.table

Suggests SIBER, gplots, fields, igraph, reshape2, rrBLUP, scales,
spelling

License GPL (>= 3)

RoxygenNote 7.3.2

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]

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

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/popula
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))) {
  res <- gl.assign.grm(platypus.gl, unknown = "T27")
}
```

---

gl.assign.mahalanobis   *Assign 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.

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, or equal to the number of dimensions with substantive eigenvalues, 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,
  dim.limit = 2,
  plevel = 0.999,
  plot.out = TRUE,
  unknown,
  verbose = NULL
)
```

Arguments

`x`	Name of the input genlight object [required].
`dim.limit`	Maximum number of dimensions to consider for the confidence ellipses [default 2]
`plevel`	Probability level for bounding ellipses [default 0.999].
`plot.out`	If TRUE, produces a plot showing the position of the unknown in relation to putative source populations [default TRUE]
`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 where no private alleles have been detected. 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 final set of putative source populations.

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.mahal() 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 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)

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

Examples

```
# Test run with a focal individual from the Macleay River (EmmacMaclGeor)
test <- gl.assign.pa(testset.gl,
  unknown = "UC_01044", nmin = 10, threshold = 1 )
test_2 <- gl.assign.pca(test, unknown = "UC_01044", plevel = 0.95)
df <- gl.assign.mahalanobis(test_2, unknown = "UC_01044")
```

---

gl.assign.pa	*Eliminates populations as possible source populations for an individual of unknown provenance, using private alleles*

Description

This script eliminates from consideration as putative source populations, those populations for which the individual has too many private alleles. The populations that remain are putative source populations, subject to further consideration.

The algorithm identifies those target populations for which the individual has no private alleles or for which the number of private alleles does not exceed a user specified threshold.

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).

Usage

```
gl.assign.pa(
  x,
  unknown,
  nmin = 10,
  threshold = 0,
  n.best = NULL,
  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].
threshold	Populations to retain for consideration; those for which the focal individual has less than or equal to threshold loci with private alleles [default 0].

| 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 [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]. |

Value

A genlight object containing the focal individual (assigned to population 'unknown') and populations for which the focal individual is not distinctive (number of loci with private alleles less than or equal to the threshold). If no such populations, the genlight object contains only data for the unknown individual.

Author(s)

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)
test <- gl.assign.pa(testset.gl,
  unknown = "UC_00146", nmin = 10, threshold = 1)
```

| gl.assign.pca | *Assign an individual of unknown provenance to population based on PCA* |

Description

This script assigns an individual of unknown provenance to one or more target populations based on its proximity to each 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 lies outside the specified confidence limits are no longer removed from the dataset.

Usage

```
gl.assign.pca(x, unknown, plevel = 0.999, 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].
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. 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 where no private alleles have been detected and the position of the unknown in relation to the confidence ellipses as is plotted by this script. Note, this plot is considering only the top two dimensions of the ordination, and so an unknown lying outside the confidence ellipse 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. 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 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)

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

Examples

```
# Test run with a focal individual from the Macleay River (EmmacMaclGeor)
test <- gl.assign.pa(testset.gl,
  unknown = "UC_00146", nmin = 10, threshold = 1,
  verbose = 3
)
```

```
test_2 <- gl.assign.pca(test, unknown = "UC_00146", plevel = 0.95, verbose = 3)
```

---

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

```
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

Two or more alleles are identical by descent (IBD) if they are identical copies of the same ancestral allele in a base population. The additive relationship matrix is a theoretical framework for estimating a relationship matrix that is consistent with an approach to estimate the probability that the alleles at a random locus are identical in state (IBS).

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

```
gl.grm(platypus.gl[1:10, 1:100])
```

gl.grm.network   *Represents a genomic relationship matrix (GRM) as a network*

Description

This script takes a G matrix generated by `gl.grm` 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.grm.network(
  G,
  x,
  method = "fr",
  node.size = 8,
  node.label = TRUE,
  node.label.size = 2,
  node.label.color = "black",
  node.shape = NULL,
  link.color = NULL,
  link.size = 2,
  relatedness_factor = 0.125,
  title = "Network based on a genomic relationship matrix",
 palette_discrete = gl.select.colors(x, library = "brewer", palette = "PuOr", ncolors =
    nPop(x), verbose = 0),
  plot.dir = NULL,
  plot.file = NULL,
  verbose = NULL
)
```

Arguments

G	A genomic relationship matrix (GRM) generated by `gl.grm` [required].
x	A genlight object from which the G matrix was generated [required].
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'].
node.shape	Optionally provide a vector of nPop shapes (run gl.select.shapes() for shape options) [default NULL].
link.color	Colors for links [default gl.select.colors].

link.size   Size of the links [default 2].

relatedness_factor
                   Factor of relatedness [default 0.125].

title   Title for the plot [default 'Network based on genomic relationship matrix'].

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) [de-
                   fault 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 using a Genomic Relationship Matrix (GRM). The GRM is generated by the
gl.grm function, which utilizes the A.mat function from the rrBLUP package. This method follows
the approach developed by Endelman and Jannink (2012).

The GRM quantifies the additive genetic relationships between individuals based on genome-wide
SNP data. It provides an estimate of the actual genetic similarity — known as realized relatedness—
between individuals by measuring how much of their genome they share identical by descent (IBD).

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.

The off-Diagonal elements of the GRM represent twice the kinship coefficient between pairs of
individuals. The kinship coefficient is the probability that a randomly selected allele from each
individual at the same locus is IBD. Diagonal elements represent one plus twice the inbreeding
coefficient of each individual. The inbreeding coefficient is the probability that both alleles at a
random locus within an individual are IBD.

Choosing meaningful thresholds to represent relationships between individuals can be challenging
because kinship and inbreeding coefficients re relative measures. To standardize the 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 indi-
viduals.

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 relatedness threshold in the function.

|Relationship|Kinship|95 |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 | - |

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)

Value

A network plot showing relatedness 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)) {
  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, relatedness_factor = 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 Place-
  ment. 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))) {
  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

```
out <- gl.report.parent.offspring(testset.gl[1:10, 1:100])
```

---

gl.run.EMIBD9 *Run program EMIBD9*

Description

Run program EMIBD9

Usage

```
gl.run.EMIBD9(
  x,
  outfile = "EMIBD9_Res.ibd9",
  outpath = tempdir(),
  emibd9.path = getwd(),
  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()].
Inbreed	A Boolean, taking values TRUE or FALSE to indicate inbreeding is not and is allowed in estimating IBD coefficients [default FALSE].
palette_convergent	
	A continuous palette function for the relatedness values [default NULL].
parallel	Use parallelisation[default FALSE].
ncores	How many cores should be used [default 1].
ISeed	An integer used to seed the random number generator [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 ($\delta 1$ - $\delta 9$) 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, $\delta 1$ to $\delta 9$ are the probabilities associated with each IBD mode. In inbreeding populations, only $\delta 1$ to $\delta 6$ can can occur. In contrast, $\delta 7$ to $\delta 9$ can only occur in large, panmictic outbred populations.

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 that could be used to guide the choosing of the relatedness threshold in the function.

|Relationship |Kinship | 95

|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].

Value

A matrix with pairwise relatedness

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
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

|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)|

|Unrelated | 0 | - |

Usage

```
gl.sim.relatedness(
  x,
  rel = "full.sib",
  nboots = 10,
  emibd9.path = getwd(),
  conf = 0.95,
  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].
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)
conf	The specified threshold for confidence interval calculation from simulated relatedness values [default 0.95]
iseed	An integer used to seed the random number generator [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]

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

## 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/popula 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")
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/popula
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")
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/popula 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")
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/popula 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")
res <- utils.assignment_2(platypus.gl, unknown = "T27")
```

Index