Package 'dartR'

December 6, 2022

Type Package

Title Importing and Analysing 'SNP' and 'Silicodart' Data Generated by Genome-Wide Restriction Fragment Analysis

Version 2.7.2

Date 2022-12-05

Description Functions are provided that facilitate the import and analysis of 'SNP' (single nucleotide polymorphism) and 'silicodart' (presence/absence) data. The main focus is on data generated by 'DarT' (Diversity Arrays Technology), however, data from other sequencing platforms can be used once 'SNP' or related fragment presence/absence data from any source is imported. Genetic datasets are stored in a derived 'genlight' format (package 'adegenet'), that allows for a very compact storage of data and metadata. Functions are available for importing and exporting of 'SNP' and 'silicodart' data, for reporting on and filtering on various criteria (e.g. 'CallRate', heterozygosity, reproducibility, maximum allele frequency). Additional functions are available for visualization (e.g. Principle Coordinate Analysis) and creating a spatial representation using maps. 'dartR' supports also the analysis of 3rd party software package such as 'newhybrid', 'structure', 'NeEstimator' and 'blast'. Since version 2.0.3 we also implemented simulation functions, that allow to forward simulate 'SNP' dynamics under different population and evolutionary dynamics. Comprehensive tutorials and support can be found at our 'github' repository: github.com/green-striped-gecko/dartR/. If you want to cite 'dartR', you find the information by typing citation('dartR') in the console.

VignetteBuilder knitr

Encoding UTF-8

Depends R (>= 3.5), adegenet (>= 2.0.0), ggplot2, dplyr, dartR.data

biocViews

Imports
ape,crayon,data.table,fields,foreach,gridExtra,MASS,methods,patchwork,plyr,PopGenReport,raster,reshape2,shiny,SNPR

Suggests boot, devtools, directlabels, dismo, doParallel, expm, gdistance, ggtern, gganimate, ggrepel, grid, gtable, ggthemes, gplots, HardyWeinberg, hierfstat, igraph, iterpc, knitr, label.switching, lattice, leaflet, leaflet.minicharts,

maptools, markdown, mmod, networkD3, parallel, pca3d, pegas, pheatmap, plotly, poppr, proxy, purrr, qvalue, RColorBrewer, Rcpp, rgl, rmarkdown, rrBLUP, scales, seqinr, shinyBS, shinyjs, shinythemes, shinyWidgets, SIBER, snpStats, stringi, terra, tibble, tidyverse, vcfR, zoo, viridis

License GPL (>= 3)

LazyData true

RoxygenNote 7.2.2

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]

Config/testthat/edition 3

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

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

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

Repository CRAN

Date/Publication 2022-12-06 10:20:02 UTC

R topics documented:

bandicoot.gl 6
cbind.dartR 7
gi2gl 7
gl.alf 8
gl.amova 9
gl.assign.grm 10
gl.assign.mahalanobis 11
gl.assign.pa 13
gl.assign.pca 14
gl.basic.stats 16
gl.blast 16
gl.check.verbosity 20
gl.collapse 20
gl.compliance.check 22
gl.costdistances 23
gl.define.pop 24
gl.diagnostics.hwe 25

gl.diagnostics.sim . . . 27
gl.dist.ind . . . 29
gl.dist.pop . . . 30
gl.drop.ind . . . 32
gl.drop.loc . . . 33
gl.drop.pop . . . 34
gl.edit.recode.ind . . . 35
gl.edit.recode.pop . . . 37
gl.evanno . . . 38
gl.fdsim . . . 39
gl.filter.allna . . . 41
gl.filter.callrate . . . 42
gl.filter.hamming . . . 44
gl.filter.heterozygosity . . . 46
gl.filter.hwe . . . 47
gl.filter.ld . . . 50
gl.filter.locmetric . . . 51
gl.filter.maf . . . 53
gl.filter.monomorphs . . . 55
gl.filter.overshoot . . . 56
gl.filter.pa . . . 57
gl.filter.parent.offspring . . . 58
gl.filter.rdepth . . . 60
gl.filter.reproducibility . . . 62
gl.filter.secondaries . . . 63
gl.filter.sexlinked . . . 64
gl.filter.taglength . . . 66
gl.fixed.diff . . . 67
gl.fst.pop . . . 69
gl.genleastcost . . . 70
gl.grm . . . 72
gl.grm.network . . . 74
gl.He . . . 77
gl.Ho . . . 77
gl.hwe.pop . . . 78
gl.ibd . . . 79
gl.impute . . . 82
gl.install.vanilla.dartR . . . 83
gl.join . . . 84
gl.keep.ind . . . 85
gl.keep.loc . . . 86
gl.keep.pop . . . 87
gl.ld.distance . . . 88
gl.ld.haplotype . . . 90
gl.LDNe . . . 92
gl.list.reports . . . 94
gl.load . . . 94
gl.make.recode.ind . . . 95

gl.make.recode.pop . . . 96
gl.map.interactive . . . 97
gl.map.structure . . . 99
gl.merge.pop . . . 101
gl.nhybrids . . . 102
gl.outflank . . . 104
gl.pcoa . . . 106
gl.pcoa.plot . . . 109
gl.percent.freq . . . 112
gl.play.history . . . 113
gl.plot.heatmap . . . 114
gl.plot.network . . . 115
gl.plot.structure . . . 117
gl.print.history . . . 119
gl.print.reports . . . 120
gl.propShared . . . 121
gl.random.snp . . . 121
gl.read.csv . . . 122
gl.read.dart . . . 124
gl.read.fasta . . . 125
gl.read.silicodart . . . 127
gl.read.vcf . . . 128
gl.reassign.pop . . . 129
gl.recalc.metrics . . . 130
gl.recode.ind . . . 131
gl.recode.pop . . . 132
gl.rename.pop . . . 133
gl.report.bases . . . 134
gl.report.callrate . . . 136
gl.report.diversity . . . 138
gl.report.hamming . . . 140
gl.report.heterozygosity . . . 142
gl.report.hwe . . . 144
gl.report.ld . . . 148
gl.report.ld.map . . . 149
gl.report.locmetric . . . 151
gl.report.maf . . . 154
gl.report.monomorphs . . . 156
gl.report.overshoot . . . 157
gl.report.pa . . . 158
gl.report.parent.offspring . . . 160
gl.report.rdepth . . . 163
gl.report.reproducibility . . . 164
gl.report.secondaries . . . 166
gl.report.sexlinked . . . 168
gl.report.taglength . . . 170
gl.run.structure . . . 172
gl.save . . . 173

gl.select.colors . . . 174
gl.select.shapes . . . 176
gl.set.verbosity . . . 177
gl.sfs . . . 178
gl.sim.create_dispersal . . . 179
gl.sim.emigration . . . 181
gl.sim.ind . . . 182
gl.sim.mutate . . . 183
gl.sim.offspring . . . 184
gl.sim.WF.run . . . 185
gl.sim.WF.table . . . 187
gl.smearplot . . . 189
gl.spatial.autoCorr . . . 190
gl.subsample.loci . . . 194
gl.test.heterozygosity . . . 195
gl.tree.nj . . . 197
gl.write.csv . . . 198
gl2bayescan . . . 199
gl2bpp . . . 200
gl2demerelate . . . 201
gl2eigenstrat . . . 202
gl2fasta . . . 203
gl2faststructure . . . 205
gl2gds . . . 206
gl2genalex . . . 207
gl2genepop . . . 208
gl2geno . . . 209
gl2gi . . . 210
gl2hiphop . . . 211
gl2phylip . . . 212
gl2plink . . . 213
gl2related . . . 215
gl2sa . . . 216
gl2sfs . . . 217
gl2shp . . . 218
gl2snapp . . . 219
gl2structure . . . 220
gl2svdquartets . . . 221
gl2treemix . . . 222
gl2vcf . . . 223
interactive_reference . . . 224
interactive_sim_run . . . 225
is.fixed . . . 225
platy . . . 226
possums.gl . . . 227
rbind.dartR . . . 227
testset.gl . . . 228
testset.gs . . . 228

testset_metadata . . . 229
testset_pop_recode . . . 229
testset_SNPs_2Row . . . 229
theme_dartR . . . 230
utils.assignment . . . 230
utils.assignment_2 . . . 231
utils.assignment_3 . . . 232
utils.assignment_4 . . . 233
utils.basic.stats . . . 234
utils.check.datatype . . . 234
utils.dart2genlight . . . 235
utils.dist.binary . . . 236
utils.dist.ind.snp . . . 238
utils.flag.start . . . 239
utils.hamming . . . 240
utils.het.pop . . . 241
utils.jackknife . . . 241
utils.n.var.invariant . . . 243
utils.outflank . . . 244
utils.outflank.MakeDiploidFSTMat . . . 246
utils.outflank.plotter . . . 246
utils.read.dart . . . 247
utils.recalc.avgpic . . . 248
utils.recalc.callrate . . . 249
utils.recalc.freqhets . . . 250
utils.recalc.freqhomref . . . 251
utils.recalc.freqhomsnp . . . 252
utils.recalc.maf . . . 253
utils.reset.flags . . . 254
utils.spautocor . . . 255
utils.structure.evanno . . . 257
utils.structure.genind2gtypes . . . 257
utils.structure.run . . . 258
zzz . . . 259
[,dartR,ANY,ANY,ANY-method . . . 259

Index **261**

`bandicoot.gl` *A genlight object created via the read.dart functions*

Description

This a test data set to test the validity of functions within dartR and is based on a DArT SNP data set of simulated bandicoots across Australia. It contains 96 individuals and 1000 SNPs.

Usage

```
bandicoot.gl
```

Format

genlight object

Author(s)

Bernd Gruber (bugs? Post to https://groups.google.com/d/forum/dartr

---

cbind.dartR *adjust cbind for dartR*

Description

cbind is a bit lazy and does not take care for the metadata (so data in the other slot is lost). You can get most of the loci metadata back using gl.compliance.check.

Usage

```
## S3 method for class 'dartR'
cbind(...)
```

Arguments

...	list of dartR objects

---

gi2gl *Converts a genind object into a genlight object*

Description

Converts a genind object into a genlight object

Usage

```
gi2gl(gi, parallel = FALSE, verbose = NULL)
```

Arguments

gi	A genind object [required].
parallel	Switch to deactivate parallel version. It might not be worth to run it parallel most of the times [default FALSE].
verbose	Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log ; 3, progress and results summary; 5, full report [default 2].

Details

Be aware due to ambiguity which one is the reference allele a combination of gi2gl(gl2gi(gl)) does not return an identical object (but in terms of analysis this conversions are equivalent)

Value

A genlight object, with all slots filled.

Author(s)

Bernd Gruber (Post to https://groups.google.com/d/forum/dartr)

---

gl.alf — *Calculates allele frequency of the first and second allele for each loci A very simple function to report allele frequencies*

Description

Calculates allele frequency of the first and second allele for each loci A very simple function to report allele frequencies

Usage

```
gl.alf(x)
```

Arguments

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

Value

A simple data.frame with alf1, alf2.

Author(s)

Bernd Gruber (bugs? Post to https://groups.google.com/d/forum/dartr)

Examples

```
#for the first 10 loci only
gl.alf(possums.gl[,1:10])
barplot(t(as.matrix(gl.alf(possums.gl[,1:10]))))
```

gl.amova *Performs AMOVA using genlight data*

Description

This script performs an AMOVA based on the genetic distance matrix from stamppNeisD() [package StAMPP] using the amova() function from the package PEGAS for exploring within and between population variation. For detailed information use their help pages: ?pegas::amova, ?StAMPP::stamppAmova. Be aware due to a conflict of the amova functions from various packages I had to 'hack' StAMPP::stamppAmova to avoid a namespace conflict.

Usage

```
gl.amova(x, distance = NULL, permutations = 100, verbose = NULL)
```

Arguments

x	Name of the genlight containing the SNP genotypes, with population information [required].
distance	Distance matrix between individuals (if not provided NeisD from StAMPP::stamppNeisD is calculated) [default NULL].
permutations	Number of permutations to perform for hypothesis testing [default 100]. Please note should be set to 1000 for analysis.
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].

Value

An object of class 'amova' which is a list with a table of sums of square deviations (SSD), mean square deviations (MSD), and the number of degrees of freedom, and a vector of variance components.

Author(s)

Bernd Gruber (bugs? Post to https://groups.google.com/d/forum/dartr)

Examples

```
#permutations should be higher, here set to 1 because of speed
out <- gl.amova(bandicoot.gl, permutations=1)
```

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)) ) {
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

```
## Not run:
#Test run with a focal individual from the Macleay River (EmmacMaclGeor)
test <- gl.assign.pa(testset.gl, unknown='UC_01044', nmin=10, threshold=1,
verbose=3)
test_2  <- gl.assign.pca(test, unknown='UC_01044', plevel=0.95, verbose=3)
df <- gl.assign.mahalanobis(test_2, unknown='UC_01044', verbose=3)

## End(Not run)
```

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,
  verbose=3)
```

---

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

```
## Not run:
#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)

## End(Not run)
```

gl.basic.stats *Calculates basic statistics for each loci (Hs, Ho, Fis etc.)*

Description

Based on function basic.stats. Check ?basic.stats for help.

Usage

```
gl.basic.stats(x, digits = 4, verbose = NULL)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
digits	Number of digits that should be returned [default 4].
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].

Value

Several tables and lists with all basic stats. basic.stats for details.

Author(s)

Bernd Gruber (bugs? Post to https://groups.google.com/d/forum/dartr)

Examples

```
if (!(requireNamespace("hierfstat", quietly = TRUE))) {
out <- gl.basic.stats(possums.gl[1:10,1:100])
}
```

gl.blast *Aligns nucleotides sequences against those present in a target database using blastn*

Description

Basic Local Alignment Search Tool (BLAST; Altschul et al., 1990 & 1997) is a sequence comparison algorithm optimized for speed used to search sequence databases for optimal local alignments to a query. This function creates fasta files, creates databases to run BLAST, runs blastn and filters these results to obtain the best hit per sequence.

This function can be used to run BLAST alignment of short-read (DArTseq data) and long-read sequences (Illumina, PacBio... etc). You can use reference genomes from NCBI, genomes from your private collection, contigs, scaffolds or any other genetic sequence that you would like to use as reference.

Usage

```
gl.blast(
  x,
  ref_genome,
  task = "megablast",
  Percentage_identity = 70,
  Percentage_overlap = 0.8,
  bitscore = 50,
  number_of_threads = 2,
  verbose = NULL
)
```

Arguments

x	Either a genlight object containing a column named 'TrimmedSequence' containing the sequence of the SNPs (the sequence tag) trimmed of adapters as provided by DArT; or a path to a fasta file with the query sequences [required].
ref_genome	Path to a reference genome in fasta of fna format [required].
task	Four different tasks are supported: 1) "megablast", for very similar sequences (e.g, sequencing errors), 2) "dc-megablast", typically used for inter-species comparisons, 3) "blastn", the traditional program used for inter-species comparisons, 4) "blastn-short", optimized for sequences less than 30 nucleotides [default 'megablast'].
Percentage_identity	Not a very sensitive or reliable measure of sequence similarity, however it is a reasonable proxy for evolutionary distance. The evolutionary distance associated with a 10 percent change in Percentage_identity is much greater at longer distances. Thus, a change from 80 – 70 percent identity might reflect divergence 200 million years earlier in time, but the change from 30 percent to 20 percent might correspond to a billion year divergence time change [default 70].
Percentage_overlap	Calculated as alignment length divided by the query length or subject length (whichever is shortest of the two lengths, i.e. length / min(qlen,slen)) [default 0.8].
bitscore	A rule-of-thumb for inferring homology, a bit score of 50 is almost always significant [default 50].
number_of_threads	Number of threads (CPUs) to use in blastn search [default 2].
verbose	verbose= 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

Installing BLAST

You can download the BLAST installs from: https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/

It is important to install BLAST in a path that does not contain spaces for this function to work.

Running BLAST

Four different tasks are supported:

- "megablast", for very similar sequences (e.g, sequencing errors)
- "dc-megablast", typically used for inter-species comparisons
- "blastn", the traditional program used for inter-species comparisons
- "blastn-short", optimized for sequences less than 30 nucleotides

If you are running a BLAST alignment of similar sequences, for example Turtle Genome Vs Turtle Sequences, the recommended parameters are: task = "megablast", Percentage_identity = 70, Percentage_overlap = 0.8 and bitscore = 50.

If you are running a BLAST alignment of highly dissimilar sequences because you are probably looking for sex linked hits in a distantly related species, and you are aligning for example sequences of Chicken Genome Vs Bassiana, the recommended parameters are: task = "dc-megablast", Percentage_identity = 50, Percentage_overlap = 0.01 and bitscore = 30.

Be aware that running BLAST might take a long time (i.e. days) depending of the size of your query, the size of your database and the number of threads selected for your computer.

BLAST output

The BLAST output is formatted as a table using output format 6, with columns defined in the following order:

- qseqid - Query Seq-id
- sacc - Subject accession
- stitle - Subject Title
- qseq - Aligned part of query sequence
- sseq - Aligned part of subject sequence
- nident - Number of identical matches
- mismatch - Number of mismatches
- pident - Percentage of identical matches
- length - Alignment length
- evalue - Expect value
- bitscore - Bit score
- qstart - Start of alignment in query
- qend - End of alignment in query
- sstart - Start of alignment in subject
- send - End of alignment in subject
- gapopen - Number of gap openings
- gaps - Total number of gaps
- qlen - Query sequence length
- slen - Subject sequence length

- PercentageOverlap - length / min(qlen,slen)

Databases containing unfiltered aligned sequences, filtered aligned sequences and one hit per sequence are saved to the temporal directory (tempdir) and can be accessed with the function gl.print.reports and listed with the function gl.list.reports. Note that they can be accessed only in the current R session because tempdir is cleared each time that the R session is closed.

BLAST filtering

BLAST output is filtered by ordering the hits of each sequence first by the highest percentage identity, then the highest percentage overlap and then the highest bitscore. Only one hit per sequence is kept based on these selection criteria.

Value

If the input is a genlight object: returns a genlight object with one hit per sequence merged to the slot $other$loc.metrics. If the input is a fasta file: returns a dataframe with one hit per sequence.

Author(s)

Berenice Talamantes Becerra & Luis Mijangos (Post to https://groups.google.com/d/forum/dartr)

References

- Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. Journal of molecular biology, 215(3), 403-410.
- Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., & Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic acids research, 25(17), 3389-3402.
- Pearson, W. R. (2013). An introduction to sequence similarity ("homology") searching. Current protocols in bioinformatics, 42(1), 3-1.

See Also

gl.print.history

Examples

```
## Not run:
res <- gl.blast(x= testset.gl,ref_genome = 'sequence.fasta')
# display of reports saved in the temporal directory
gl.list.reports()
# open the reports saved in the temporal directory
blast_databases <- gl.print.reports(1)

## End(Not run)
```

gl.check.verbosity *Checks the current global verbosity*

Description

The verbosity can be set in one of two ways – (a) explicitly by the user by passing a value using the parameter verbose= in a function, or (b) by setting the verbosity globally as part of the r environment (gl.set.verbosity).

Usage

```
gl.check.verbosity(x = NULL)
```

Arguments

x	User requested level of verbosity [default NULL].

Value

The verbosity, in variable verbose

Author(s)

Bernd Gruber (Post to https://groups.google.com/d/forum/dartr)

gl.collapse *Collapses a distance matrix by amalgamating populations with pairwise fixed difference count less that a threshold*

Description

This script takes a file generated by gl.fixed.diff and amalgamates populations with distance less than or equal to a specified threshold. The distance matrix is generated by gl.fixed.diff().

The script then applies the new population assignments to the genlight object and recalculates the distance and associated matrices.

Usage

```
gl.collapse(fd, tpop = 0, tloc = 0, pb = FALSE, verbose = NULL)
```

Arguments

fd	Name of the list of matrices produced by gl.fixed.diff() [required].
tpop	Threshold number of fixed differences above which populations will not be amalgamated [default 0].
tloc	Threshold defining a fixed difference (e.g. 0.05 implies 95:5 vs 5:95 is fixed) [default 0].
pb	If TRUE, show a progress bar on time consuming loops [default FALSE].
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 list containing the gl object x and the following square matrices:

1. $gl – the new genlight object with populations collapsed;
2. $fd – raw fixed differences;
3. $pcfd – percent fixed differences;
4. $nobs – mean no. of individuals used in each comparison;
5. $nloc – total number of loci used in each comparison;
6. $expfpos – NA's, populated by gl.fixed.diff [by simulation]
7. $expfpos – NA's, populated by gl.fixed.diff [by simulation]
8. $prob – NA's, populated by gl.fixed.diff [by simulation]

Author(s)

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

Examples

```
fd <- gl.fixed.diff(testset.gl,tloc=0.05)
fd
fd2 <- gl.collapse(fd,tpop=1)
fd2
fd3 <- gl.collapse(fd2,tpop=1)
fd3

 fd <- gl.fixed.diff(testset.gl,tloc=0.05)
 fd2 <- gl.collapse(fd)
```

`gl.compliance.check` *Checks a genlight object to see if it complies with dartR expectations and amends it to comply if necessary*

Description

This function will check to see that the genlight object conforms to expectation in regard to dartR requirements (see details), and if it does not, will rectify it.

Usage

```
gl.compliance.check(x, verbose = NULL)
```

Arguments

`x`	Name of the input genlight object [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

A genlight object used by dartR has a number of requirements that allow functions within the package to operate correctly. The genlight object comprises:

1. The SNP genotypes or Tag Presence/Absence data (SilicoDArT);
2. An associated dataframe (gl@other$loc.metrics) containing the locus metrics (e.g. Call Rate, Repeatability, etc);
3. An associated dataframe (gl@other$ind.metrics) containing the individual/sample metrics (e.g. sex, latitude (=lat), longitude(=lon), etc);
4. A specimen identity field (indNames(gl)) with the unique labels applied to each individual/sample;
5. A population assignment (popNames) for each individual/specimen;
6. Flags that indicate whether or not calculable locus metrics have been updated.

Value

A genlight object that conforms to the expectations of dartR

Author(s)

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

Examples

```
x <- gl.compliance.check(testset.gl)
x <- gl.compliance.check(testset.gs)
```

gl.costdistances *Calculates cost distances for a given landscape (resistance matrix)*

Description

Calculates a cost distance matrix, to be used with run.popgensim.

Usage

```
gl.costdistances(landscape, locs, method, NN, verbose = NULL)
```

Arguments

landscape	A raster object coding the resistance of the landscape [required].
locs	Coordinates of the subpopulations. If a genlight object is provided coordinates are taken from @other$latlon and centers for population (pop(gl)) are calculated. In case you want to calculate costdistances between individuals redefine pop(gl) via: pop(gl)<- indNames(gl) [required].
method	Defines the type of cost distance, types are 'leastcost', 'rSPDistance' or 'commute' (Circuitscape type) [required].
NN	Number of next neighbours recommendation is 8 [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].

Value

A costdistance matrix between all pairs of locs.

Examples

```
## Not run:
data(possums.gl)
library(raster)  #needed for that example
landscape.sim <- readRDS(system.file('extdata','landscape.sim.rdata',
package='dartR'))
#calculate mean centers of individuals per population
xy <- apply(possums.gl@other$xy, 2, function(x) tapply(x, pop(possums.gl),
 mean))
cd <- gl.costdistances(landscape.sim, xy, method='leastcost', NN=8)
round(cd,3)

## End(Not run)
```

`gl.define.pop` *Defines a new population in a genlight object for specified individuals*

Description

The script reassigns existing individuals to a new population and removes their existing population assignment.

The script returns a genlight object with the new population assignment.

Usage

```
gl.define.pop(x, ind.list, new, verbose = NULL)
```

Arguments

`x`	Name of the genlight object containing SNP genotypes [required].
`ind.list`	A list of individuals to be assigned to the new population [required].
`new`	Name of the new 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 or as specified using gl.set.verbosity].

Value

A genlight object with the redefined population structure.

Author(s)

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

Examples

```
popNames(testset.gl)
gl <- gl.define.pop(testset.gl, ind.list=c('AA019073','AA004859'),
new='newguys')
popNames(gl)
indNames(gl)[pop(gl)=='newguys']
```

`gl.diagnostics.hwe` *Provides descriptive stats and plots to diagnose potential problems with Hardy-Weinberg proportions*

Description

Different causes may be responsible for lack of Hardy-Weinberg proportions. This function helps diagnose potential problems.

Usage

```
gl.diagnostics.hwe(
  x,
  alpha_val = 0.05,
  bins = 20,
  stdErr = TRUE,
  colors_hist = two_colors,
  colors_barplot = two_colors_contrast,
  plot_theme = theme_dartR(),
  save2tmp = FALSE,
  n.cores = "auto",
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object containing the SNP data [required].
`alpha_val`	Level of significance for testing [default 0.05].
`bins`	Number of bins to display in histograms [default 20].
`stdErr`	Whether standard errors for Fis and Fst should be computed (default: TRUE)
`colors_hist`	List of two color names for the borders and fill of the histogram [default two_colors].
`colors_barplot`	Vector with two color names for the observed and expected number of significant HWE tests [default two_colors_contrast].
`plot_theme`	User specified theme [default theme_dartR()].
`save2tmp`	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
`n.cores`	The number of cores to use. If "auto", it will use all but one available cores [default "auto"].
`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

This function initially runs `gl.report.hwe` and reports the ternary plots. The remaining outputs follow the recommendations from Waples (2015) paper and De Meeûs 2018. These include:

1. A histogram with the distribution of p-values of the HWE tests. The distribution should be roughly uniform across equal-sized bins.
2. A bar plot with observed and expected (null expectation) number of significant HWE tests for the same locus in multiple populations (that is, the x-axis shows whether a locus results significant in 1, 2, ..., n populations. The y axis is the count of these occurrences. The zero value on x-axis shows the number of non-significant tests). If HWE tests are significant by chance alone, observed and expected number of HWE tests should have roughly a similar distribution.
3. A scatter plot with a linear regression between Fst and Fis, averaged across subpopulations. De Meeûs 2018 suggests that in the case of Null alleles, a strong positive relationship is expected (together with the Fis standard error much larger than the Fst standard error, see below). **Note**, this is not the scatter plot that Waples 2015 presents in his paper. In the lower right corner of the plot, the Pearson correlation coefficient is reported.
4. The Fis and Fst (averaged over loci and subpopulations) standard errors are also printed on screen and reported in the returned list (if `stdErr=TRUE`). These are computed with the Jackknife method over loci (See De Meeûs 2007 for details on how this is computed) and it may take some time for these computations to complete. De Meeûs 2018 suggests that under a global significant heterozygosity deficit:

   - if the correlation between Fis and Fst is strongly positive, and StdErrFis » StdErrFst, Null alleles are likely to be the cause.

   - if the correlation between Fis and Fst is ~0 or mildly positive, and StdErrFis > StdErrFst, Wahlund may be the cause.

   - if the correlation between Fis and Fst is ~0, and StdErrFis ~ StdErrFst, selfing or sib mating could to be the cause.

   It is important to realise that these statistics only suggest a pattern (pointers). Their absence is not conclusive evidence of the absence of the problem, as their presence does not confirm the cause of the problem.
5. A table where the number of observed and expected significant HWE tests are reported by each population, indicating whether these are due to heterozygosity excess or deficiency. These can be used to have a clue of potential problems (e.g. deficiency might be due to a Wahlund effect, presence of null alleles or non-random sampling; excess might be due to sex linkage or different selection between sexes, demographic changes or small Ne. See Table 1 in Wapples 2015). The last two columns of the table generated by this function report chisquare values and their associated p-values. Chisquare is computed following Fisher's procedure for a global test (Fisher 1970). This basically tests whether there is at least one test that is truly significant in the series of tests conducted (De Meeûs et al 2009).

Value

A list with the table with the summary of the HWE tests and (if stdErr=TRUE) a named vector with the StdErrFis and StdErrFst.

Author(s)

Custodian: Carlo Pacioni – Post to https://groups.google.com/d/forum/dartr

References

- de Meeûs, T., McCoy, K.D., Prugnolle, F., Chevillon, C., Durand, P., Hurtrez-Boussès, S., Renaud, F., 2007. Population genetics and molecular epidemiology or how to "débusquer la bête". Infection, Genetics and Evolution 7, 308-332.
- De Meeûs, T., Guégan, J.-F., Teriokhin, A.T., 2009. MultiTest V.1.2, a program to binomially combine independent tests and performance comparison with other related methods on proportional data. BMC Bioinformatics 10, 443-443.
- De Meeûs, T., 2018. Revisiting FIS, FST, Wahlund Effects, and Null Alleles. Journal of Heredity 109, 446-456.
- Fisher, R., 1970. Statistical methods for research workers Edinburgh: Oliver and Boyd.
- Waples, R. S. (2015). Testing for Hardy–Weinberg proportions: have we lost the plot?. Journal of heredity, 106(1), 1-19.

See Also

gl.report.hwe

Examples

```
## Not run:
require("dartR.data")
res <- gl.diagnostics.hwe(x = gl.filter.allna(platypus.gl[,1:50]),
stdErr=FALSE, n.cores=1)

## End(Not run)
```

---

gl.diagnostics.sim *Comparing simulations against theoretical expectations*

Description

Comparing simulations against theoretical expectations

Usage

```
gl.diagnostics.sim(
  x,
  Ne,
  iteration = 1,
  pop_he = 1,
  pops_fst = c(1, 2),
  plot_theme = theme_dartR(),
```

```
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Output from function gl.sim.WF.run [required].
Ne	Effective population size to use as input to compare theoretical expectations [required].
iteration	Iteration number to analyse [default 1].
pop_he	Population name in which the rate of loss of heterozygosity is going to be compared against theoretical expectations [default 1].
pops_fst	Pair of populations in which FST is going to be compared against theoretical expectations [default c(1,2)].
plot_theme	User specified theme [default theme_dartR()].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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

Two plots are presented comparing the simulations against theoretical expectations:

1. Expected heterozygosity under neutrality (Crow & Kimura, 1970, p. 329) is calculated as:
   Het = He0(1-(1/2Ne))^t,
   where Ne is effective population size, He0 is heterozygosity at generation 0 and t is the number of generations.
2. Expected FST under neutrality (Takahata, 1983) is calculated as:
   FST=1/(4Nem(n/(n-1))^2+1),
   where Ne is effective populations size of each individual subpopulation, m is dispersal rate and n the number of subpopulations (always 2).

Value

Returns plots comparing simulations against theoretical expectations

Author(s)

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

References

- Crow JF, Kimura M. An introduction to population genetics theory. An introduction to population genetics theory. 1970.
- Takahata N. Gene identity and genetic differentiation of populations in the finite island model. Genetics. 1983;104(3):497-512.

See Also

gl.filter.callrate

---

gl.dist.ind *Calculates a distance matrix for individuals defined in a genlight object*

Description

This script calculates various distances between individuals based on allele frequencies or presence-absence data

Usage

```
gl.dist.ind(
  x,
  method = NULL,
  scale = FALSE,
  swap = FALSE,
  output = "dist",
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = two_colors,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight containing the SNP genotypes or presence-absence data [required].
method	Specify distance measure [SNP: Euclidean; P/A: Simple].
scale	If TRUE, the distances are scaled to fall in the range [0,1] [default TRUE]
swap	If TRUE and working with presence-absence data, then presence (no disrupting mutation) is scored as 0 and absence (presence of a disrupting mutation) is scored as 1 [default FALSE].
output	Specify the format and class of the object to be returned, 'dist' for a object of class dist, 'matrix' for an object of class matrix [default "dist"].
plot.out	If TRUE, display a histogram and a boxplot of the genetic distances [TRUE].
plot_theme	User specified theme [default theme_dartR].
plot_colors	Vector with two color names for the borders and fill [default two_colors].
save2tmp	If TRUE, saves any ggplots to the session temporary directory [default FALSE].
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 distance measure for SNP genotypes can be one of:

- Euclidean Distance [method = "Euclidean"]
- Scaled Euclidean Distance [method='Euclidean", scale=TRUE]
- Simple Mismatch Distance [method="Simple"]
- Absolute Mismatch Distance [method="Absolute"]
- Czekanowski (Manhattan) Distance [method="Manhattan"]

The distance measure for Sequence Tag Presence/Absence data (binary) can be one of:

- Euclidean Distance [method = "Euclidean"]
- Scaled Euclidean Distance [method='Euclidean", scale=TRUE]
- Simple Matching Distance [method="Simple"]
- Jaccard Distance [method="Jaccard"]
- Bray-Curtis Distance [method="Bray-Curtis"]

Refer to the dartR Technical Note on Distances in Genetics.

Value

An object of class 'matrix' or dist' giving distances between individuals

Author(s)

Author(s): Arthur Georges. Custodian: Arthur Georges – Post to #' https://groups.google.com/d/forum/dartr

Examples

```
D <- gl.dist.ind(testset.gl[1:20,], method='manhattan')
D <- gl.dist.ind(testset.gs[1:20,], method='Jaccard',swap=TRUE)

D <- gl.dist.ind(testset.gl[1:20,], method='euclidean',scale=TRUE)
```

---

gl.dist.pop *Calculates a distance matrix for populations with SNP genotypes in a genlight object*

Description

This script calculates various distances between populations based on allele frequencies (SNP genotypes) or frequency of presences in presence-absence data (Euclidean and Fixed-diff distances only).

Usage

```
gl.dist.pop(
  x,
  method = "euclidean",
  plot.out = TRUE,
  scale = FALSE,
  output = "dist",
  plot_theme = theme_dartR(),
  plot_colors = two_colors,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight containing the SNP genotypes [required].
method	Specify distance measure [default euclidean].
plot.out	If TRUE, display a histogram of the genetic distances, and a whisker plot [default TRUE].
scale	If TRUE and method='Euclidean', the distance will be scaled to fall in the range [0,1] [default FALSE].
output	Specify the format and class of the object to be returned, dist for a object of class dist, matrix for an object of class matrix [default "dist"].
plot_theme	User specified theme [default theme_dartR()].
plot_colors	Vector with two color names for the borders and fill [default two_colors].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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 distance measure can be one of 'euclidean', 'fixed-diff', 'reynolds', 'nei' and 'chord'. Refer to the documentation of functions described in the the dartR Distance Analysis tutorial for algorithms and definitions.

Value

An object of class 'dist' giving distances between populations

Author(s)

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

Examples

```
 ## Not run:
# SNP genotypes
D <- gl.dist.pop(possums.gl[1:90,1:100], method='euclidean')
D <- gl.dist.pop(possums.gl[1:90,1:100], method='euclidean',scale=TRUE)
#D <- gl.dist.pop(possums.gl, method='nei')
#D <- gl.dist.pop(possums.gl, method='reynolds')
#D <- gl.dist.pop(possums.gl, method='chord')
#D <- gl.dist.pop(possums.gl, method='fixed-diff')
#Presence-Absence data [only 10 individuals due to speed]
D <- gl.dist.pop(testset.gs[1:10,], method='euclidean')

## End(Not run)
res <- gl.dist.pop(platypus.gl)
```

---

gl.drop.ind *Removes specified individuals from a genlight {adegenet} object*

Description

The script, having deleted individuals, optionally identifies resultant monomorphic loci or loci with all values missing and deletes them (using gl.filter.monomorphs.r). The script also optionally recalculates statistics made redundant by the deletion of individuals from the dataset.

The script returns a genlight object with the individuals deleted and, optionally, the recalculated locus metadata.

Usage

```
gl.drop.ind(x, ind.list, recalc = FALSE, mono.rm = FALSE, verbose = NULL)
```

Arguments

x	Name of the genlight object containing SNP genotypes [required].
ind.list	A list of individuals to be removed [required].
recalc	Recalculate the locus metadata statistics [default FALSE].
mono.rm	Remove monomorphic loci [default FALSE].
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 with the reduced data

Author(s)

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

See Also

gl.keep.ind to keep rather than drop specified individuals

Examples

```
# SNP data
  gl2 <- gl.drop.ind(testset.gl,
  ind.list=c('AA019073','AA004859'))
  gl2 <- gl.drop.ind(testset.gl,
  ind.list=c('AA019073','AA004859'))
# Tag P/A data
  gs2 <- gl.drop.ind(testset.gs,
  ind.list=c('AA020656','AA19077','AA004859'))
  gs2 <- gl.drop.ind(testset.gs, ind.list=c('AA020656'
  ,'AA19077','AA004859'),mono.rm=TRUE, recalc=TRUE)
```

---

gl.drop.loc *Removes specified loci from a genlight {adegenet} object*

Description

The script returns a genlight object with specified loci deleted.

Usage

```
gl.drop.loc(x, loc.list = NULL, first = NULL, last = NULL, verbose = NULL)
```

Arguments

x	Name of the genlight object containing SNP genotypes or presence/absence data [required].
loc.list	A list of loci to be deleted [required, if loc.range not specified].
first	First of a range of loci to be deleted [required, if loc.list not specified].
last	Last of a range of loci to be deleted [if not specified, last locus in the dataset].
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 with the reduced data.

Author(s)

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

See Also

gl.keep.loc to keep rather than drop specified loci

Examples

```
# SNP data
 gl2 <- gl.drop.loc(testset.gl, loc.list=c('100051468|42-A/T', '100049816-51-A/G'),verbose=3)
# Tag P/A data
  gs2 <- gl.drop.loc(testset.gs, loc.list=c('20134188','19249144'),verbose=3)
```

---

gl.drop.pop *Removes specified populations from a genlight object*

Description

Individuals are assigned to populations based on the specimen metadata file (csv) used with gl.read.dart. The script, having deleted populations, optionally identifies resultant monomorphic loci or loci with all values missing and deletes them (using gl.filter.monomorphs.r). The script also optionally recalculates statistics made redundant by the deletion of individuals from the dataset.

The script returns a genlight object with the new population assignments and the recalculated locus metadata.

Usage

```
gl.drop.pop(
  x,
  pop.list,
  as.pop = NULL,
  recalc = FALSE,
  mono.rm = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing SNP genotypes or Tag P/A data (SilicoDArT) [required].
pop.list	A list of populations to be removed [required].
as.pop	Temporarily assign another metric to represent population for the purposes of deletions [default NULL].
recalc	Recalculate the locus metadata statistics [default FALSE].
mono.rm	Remove monomorphic loci [default FALSE].
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 with the reduced data

Author(s)

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

See Also

gl.keep.pop to keep rather than drop specified populations

Examples

```
# SNP data
  gl2 <- gl.drop.pop(testset.gl,
  pop.list=c('EmsubRopeMata','EmvicVictJasp'),verbose=3)
  gl2 <- gl.drop.pop(testset.gl, pop.list=c('EmsubRopeMata','EmvicVictJasp'),
  mono.rm=TRUE,recalc=TRUE)
  gl2 <- gl.drop.pop(testset.gl, pop.list=c('Male','Unknown'),as.pop='sex')
# Tag P/A data
  gs2 <- gl.drop.pop(testset.gs, pop.list=c('EmsubRopeMata','EmvicVictJasp'))
```

---

gl.edit.recode.ind *Creates or edits individual (=specimen) names, creates a recode_ind file and applies the changes to a genlight object*

Description

A script to edit individual names in a genlight object, or to create a reassignment table taking the individual labels from a genlight object, or to edit existing individual labels in an existing recode_ind file.

Usage

```
gl.edit.recode.ind(
  x,
  out.recode.file = NULL,
  outpath = tempdir(),
  recalc = FALSE,
  mono.rm = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object for which individuals are to be relabelled [required].
out.recode.file	Name of the file to output the new individual labels [optional].
outpath	Path where to save the output file [default tempdir(), mandated by CRAN].
recalc	Recalculate the locus metadata statistics [default TRUE].
mono.rm	Remove monomorphic loci [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

Renaming individuals may be required when there have been errors in labelling arising in the process from sample to DArT files. There may be occasions where renaming individuals is required for preparation of figures. Caution needs to be exercised because of the potential for breaking the 'chain of evidence' between the samples themselves and the analyses. Recoding individuals can also be done with a recode table (csv).

This script will input an existing recode table for editing and optionally save it as a new table, or if the name of an input table is not supplied, will generate a table using the individual labels in the parent genlight object.

The script, having deleted individuals, optionally identifies resultant monomorphic loci or loci with all values missing and deletes them (using gl.filter.monomorphs.r). The script also optionally recalculates statistics made redundant by the deletion of individuals from the dataset.

Use outpath=getwd() or outpath='.' when calling this function to direct output files to your working directory.

The script returns a genlight object with the new individual labels and the recalculated locus metadata.

Value

An object of class ('genlight') with the revised individual labels.

Author(s)

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

See Also

gl.recode.ind, gl.drop.ind, gl.keep.ind

Examples

```
## Not run:
gl <- gl.edit.recode.ind(testset.gl)
gl <- gl.edit.recode.ind(testset.gl, out.recode.file='ind.recode.table.csv')

## End(Not run)
```

`gl.edit.recode.pop` *Creates or edits a population re-assignment table*

Description

A script to edit population assignments in a genlight object, or to create a reassignment table taking the population assignments from a genlight object, or to edit existing population assignments in a pop.recode.table.

Usage

```
gl.edit.recode.pop(
  x,
  pop.recode = NULL,
  out.recode.file = NULL,
  outpath = tempdir(),
  recalc = FALSE,
  mono.rm = FALSE,
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object for which populations are to be reassigned [required].
`pop.recode`	Path to recode file [default NULL].
`out.recode.file`	Name of the file to output the new individual labels [default NULL].
`outpath`	Path where to save the output file [default tempdir(), mandated by CRAN].
`recalc`	Recalculate the locus metadata statistics if any individuals are deleted [default TRUE].
`mono.rm`	Remove monomorphic loci [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

Genlight objects assign specimens to populations based on information in the ind.metadata file provided when the genlight object is first generated. Often one wishes to subset the data by deleting populations or to amalgamate populations. This can be done with a pop.recode table with two columns. The first column is the population assignment in the genlight object, the second column provides the new assignment.

This script will input an existing reassignment table for editting and optionally save it as a new table, or if the name of an input table is not supplied, will generate a table using the population assignments in the parent genlight object.

The script, having deleted populations, optionally identifies resultant monomorphic loci or loci with all values missing and deletes them (using gl.filter.monomorphs.r). The script also optionally recalculates statistics made redundant by the deletion of individuals from the dataset.

Use outpath=getwd() or outpath='.' when calling this function to direct output files to your working directory.

The script returns a genlight object with the new population assignments and the recalculated locus metadata.

Value

An object of class ('genlight') with the revised population assignments

Author(s)

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

See Also

gl.recode.pop, gl.drop.pop, gl.keep.pop, gl.merge.pop, gl.reassign.pop

Examples

```
## Not run:
gl <- gl.edit.recode.pop(testset.gl)
gs <- gl.edit.recode.pop(testset.gs)

## End(Not run)
```

---

gl.evanno *Creates an Evanno plot from a STRUCTURE run object*

Description

This function takes a genlight object and runs a STRUCTURE analysis based on functions from strataG

Usage

```
gl.evanno(sr, plot.out = TRUE)
```

Arguments

sr	structure run object from gl.run.structure [required].
plot.out	TRUE: all four plots are shown. FALSE: all four plots are returned as a ggplot but not shown [default TRUE].

Details

The function is basically a convenient wrapper around the beautiful strataG function evanno (Archer et al. 2016). For a detailed description please refer to this package (see references below).

Value

An Evanno plot is created and a list of all four plots is returned.

Author(s)

Bernd Gruber (Post to https://groups.google.com/d/forum/dartr)

References

- Pritchard, J.K., Stephens, M., Donnelly, P. (2000) Inference of population structure using multilocus genotype data. Genetics 155, 945-959.
- Archer, F. I., Adams, P. E. and Schneiders, B. B. (2016) strataG: An R package for manipulating, summarizing and analysing population genetic data. Mol Ecol Resour. doi:10.1111/1755-0998.12559
- Evanno, G., Regnaut, S., and J. Goudet. 2005. Detecting the number of clusters of individuals using the software STRUCTURE: a simulation study. Molecular Ecology 14:2611-2620.

See Also

gl.run.structure, clumpp,

Examples

```
## Not run:
#CLUMPP and STRUCTURE need to be installed to be able to run the example
#bc <- bandicoot.gl[,1:100]
#sr <- gl.run.structure(bc, k.range = 2:5, num.k.rep = 3, exec = './structure.exe')
#ev <- gl.evanno(sr)
#ev
#qmat <- gl.plot.structure(sr, k=3, CLUMPP='d:/structure/')
#head(qmat)
#gl.map.structure(qmat, bc, scalex=1, scaley=0.5)

## End(Not run)
```

---

gl.fdsim *Estimates the rate of false positives in a fixed difference analysis*

Description

This function takes two populations and generates allele frequency profiles for them. It then samples an allele frequency for each, at random, and estimates a sampling distribution for those two allele frequencies. Drawing two samples from those sampling distributions, it calculates whether or not they represent a fixed difference. This is applied to all loci, and the number of fixed differences so generated are counted, as an expectation. The script distinguished between true fixed differences (with a tolerance of delta), and false positives. The simulation is repeated a given number of times (default=1000) to provide an expectation of the number of false positives, given the observed allele frequency profiles and the sample sizes. The probability of the observed count of fixed differences is greater than the expected number of false positives is calculated.

Usage

```
gl.fdsim(
  x,
  poppair,
  obs = NULL,
  sympatric = FALSE,
  reps = 1000,
  delta = 0.02,
  verbose = NULL
)
```

Arguments

x	Name of the genlight containing the SNP genotypes [required].
poppair	Labels of two populations for comparison in the form c(popA,popB) [required].
obs	Observed number of fixed differences between the two populations [default NULL].
sympatric	If TRUE, the two populations are sympatric, if FALSE then allopatric [default FALSE].
reps	Number of replications to undertake in the simulation [default 1000].
delta	The threshold value for the minor allele frequency to regard the difference between two populations to be fixed [default 0.02].
verbose	Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2].

Value

A list containing the following square matrices [[1]] observed fixed differences; [[2]] mean expected number of false positives for each comparison; [[3]] standard deviation of the no. of false positives for each comparison; [[4]] probability the observed fixed differences arose by chance for each comparison.

Author(s)

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

Examples

```
fd <- gl.fdsim(testset.gl[,1:100],poppair=c('EmsubRopeMata','EmmacBurnBara'),
sympatric=TRUE,verbose=3)
```

`gl.filter.allna` *Filters loci that are all NA across individuals and/or populations with all NA across loci*

Description

This script deletes deletes loci or individuals with all calls missing (NA), from a genlight object

A DArT dataset will not have loci for which the calls are scored all as missing (NA) for a particular individual, but such loci can arise rarely when populations or individuals are deleted. Similarly, a DArT dataset will not have individuals for which the calls are scored all as missing (NA) across all loci, but such individuals may sneak in to the dataset when loci are deleted. Retaining individual or loci with all NAs can cause issues for several functions.

Also, on occasion an analysis will require that there are some loci scored in each population. Setting by.pop=TRUE will result in removal of loci when they are all missing in any one population.

Note that loci that are missing for all individuals in a population are not imputed with method 'frequency' or 'HW'. Consider using the function `gl.filter.allna` with by.pop=TRUE.

Usage

```
gl.filter.allna(x, by.pop = FALSE, recalc = FALSE, verbose = NULL)
```

Arguments

`x`	Name of the input genlight object [required].
`by.pop`	If TRUE, loci that are all missing in any one population are deleted [default FALSE]
`recalc`	Recalculate the locus metadata statistics if any individuals are deleted in the filtering [default FALSE].
`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].

Value

A genlight object having removed individuals that are scored NA across all loci, or loci that are scored NA across all individuals.

Author(s)

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

See Also

Other filter functions: gl.filter.callrate(), gl.filter.heterozygosity(), gl.filter.hwe(), gl.filter.ld(), gl.filter.locmetric(), gl.filter.maf(), gl.filter.monomorphs(), gl.filter.overshoot(), gl.filter.parent.offspring(), gl.filter.pa(), gl.filter.rdepth(), gl.filter.reproducibility(), gl.filter.secondaries(), gl.filter.sexlinked(), gl.filter.taglength()

Examples

```
# SNP data
  result <- gl.filter.allna(testset.gl, verbose=3)
# Tag P/A data
  result <- gl.filter.allna(testset.gs, verbose=3)
```

---

gl.filter.callrate *Filters loci or specimens in a genlight {adegenet} object based on call rate*

Description

SNP datasets generated by DArT have missing values primarily arising from failure to call a SNP because of a mutation at one or both of the restriction enzyme recognition sites. The script gl.filter.callrate() will filter out the loci with call rates below a specified threshold.

Tag Presence/Absence datasets (SilicoDArT) have missing values where it is not possible to determine reliably if there the sequence tag can be called at a particular locus.

Usage

```
gl.filter.callrate(
  x,
  method = "loc",
  threshold = 0.95,
  mono.rm = FALSE,
  recalc = FALSE,
  recursive = FALSE,
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = two_colors,
  bins = 25,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data, or the genind object containing the SilocoDArT data [required].
method	Use method='loc' to specify that loci are to be filtered, 'ind' to specify that specimens are to be filtered, 'pop' to remove loci that fail to meet the specified threshold in any one population [default 'loc'].
threshold	Threshold value below which loci will be removed [default 0.95].
mono.rm	Remove monomorphic loci after analysis is complete [default FALSE].
recalc	Recalculate the locus metadata statistics if any individuals are deleted in the filtering [default FALSE].
recursive	Repeatedly filter individuals on call rate, each time removing monomorphic loci. Only applies if method='ind' and mono.rm=TRUE [default FALSE].
plot.out	Specify if histograms of call rate, before and after, are to be produced [default TRUE].
plot_theme	User specified theme for the plot [default theme_dartR()].
plot_colors	List of two color names for the borders and fill of the plots [default two_colors].
bins	Number of bins to display in histograms [default 25].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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

Because this filter operates on call rate, this function recalculates Call Rate, if necessary, before filtering. If individuals are removed using method='ind', then the call rate stored in the genlight object is, optionally, recalculated after filtering.

Note that when filtering individuals on call rate, the initial call rate is calculated and compared against the threshold. After filtering, if mono.rm=TRUE, the removal of monomorphic loci will alter the call rates. Some individuals with a call rate initially greater than the nominated threshold, and so retained, may come to have a call rate lower than the threshold. If this is a problem, repeated iterations of this function will resolve the issue. This is done by setting mono.rm=TRUE and recursive=TRUE, or it can be done manually.

Callrate is summarized by locus or by individual to allow sensible decisions on thresholds for filtering taking into consideration consequential loss of data. The summary is in the form of a tabulation and plots.

Plot themes can be obtained from

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

Resultant ggplot(s) and the tabulation(s) are saved to the session's temporary directory.

Value

The reduced genlight or genind object, plus a summary

Author(s)

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

See Also

gl.report.callrate

Other filter functions: gl.filter.allna(), gl.filter.heterozygosity(), gl.filter.hwe(), gl.filter.ld(), gl.filter.locmetric(), gl.filter.maf(), gl.filter.monomorphs(), gl.filter.overshoot(), gl.filter.parent.offspring(), gl.filter.pa(), gl.filter.rdepth(), gl.filter.reproducibility(), gl.filter.secondaries(), gl.filter.sexlinked(), gl.filter.taglength()

Examples

```
# SNP data
  result <- gl.filter.callrate(testset.gl[1:10], method='loc', threshold=0.8,
   verbose=3)
  result <- gl.filter.callrate(testset.gl[1:10], method='ind', threshold=0.8,
   verbose=3)
  result <- gl.filter.callrate(testset.gl[1:10], method='pop', threshold=0.8,
   verbose=3)
# Tag P/A data
  result <- gl.filter.callrate(testset.gs[1:10], method='loc',
  threshold=0.95, verbose=3)
  result <- gl.filter.callrate(testset.gs[1:10], method='ind',
  threshold=0.8, verbose=3)
  result <- gl.filter.callrate(testset.gs[1:10], method='pop',
  threshold=0.8, verbose=3)

  res <- gl.filter.callrate(platypus.gl)
```

---

gl.filter.hamming *Filters loci based on pairwise Hamming distance between sequence tags*

Description

Hamming distance is calculated as the number of base differences between two sequences which can be expressed as a count or a proportion. Typically, it is calculated between two sequences of equal length. In the context of DArT trimmed sequences, which differ in length but which are anchored to the left by the restriction enzyme recognition sequence, it is sensible to compare the two trimmed sequences starting from immediately after the common recognition sequence and terminating at the last base of the shorter sequence.

Usage

```
gl.filter.hamming(
  x,
  threshold = 0.2,
  rs = 5,
  taglength = 69,
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = two_colors,
  pb = FALSE,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
threshold	A threshold Hamming distance for filtering loci [default threshold 0.2].
rs	Number of bases in the restriction enzyme recognition sequence [default 5].
taglength	Typical length of the sequence tags [default 69].
plot.out	Specify if plot is to be produced [default TRUE].
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 two_colors].
pb	Switch to output progress bar [default FALSE].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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

Hamming distance can be computed by exploiting the fact that the dot product of two binary vectors x and (1-y) counts the corresponding elements that are different between x and y. This approach can also be used for vectors that contain more than two possible values at each position (e.g. A, C, T or G).

If a pair of DNA sequences are of differing length, the longer is truncated.

The algorithm is that of Johann de Jong https://johanndejong.wordpress.com/2015/10/02/faster-hamming-distance-in-r-2/ as implemented in utils.hamming.

Only one of two loci are retained if their Hamming distance is less that a specified percentage. 5 base differences out of 100 bases is a 20

Value

A genlight object filtered on Hamming distance.

Author(s)

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

Examples

```
# SNP data
test <- platypus.gl
test <- gl.subsample.loci(platypus.gl,n=50)
result <- gl.filter.hamming(test, threshold=0.25, verbose=3)
```

---

gl.filter.heterozygosity

Filters individuals with average heterozygosity greater than a specified upper threshold or less than a specified lower threshold

---

Description

Calculates the observed heterozygosity for each individual in a genlight object and filters individuals based on specified threshold values. Use gl.report.heterozygosity to determine the appropriate thresholds.

Usage

```
gl.filter.heterozygosity(x, t.upper = 0.7, t.lower = 0, verbose = NULL)
```

Arguments

x	A genlight object containing the SNP genotypes [required].
t.upper	Filter individuals > the threshold [default 0.7].
t.lower	Filter individuals < the threshold [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, unless specified using gl.set.verbosity].

Value

The filtered genlight object.

Author(s)

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

See Also

Other filter functions: gl.filter.allna(), gl.filter.callrate(), gl.filter.hwe(), gl.filter.ld(), gl.filter.locmetric(), gl.filter.maf(), gl.filter.monomorphs(), gl.filter.overshoot(), gl.filter.parent.offspring(), gl.filter.pa(), gl.filter.rdepth(), gl.filter.reproducibility(), gl.filter.secondaries(), gl.filter.sexlinked(), gl.filter.taglength()

Examples

```
result <- gl.filter.heterozygosity(testset.gl,t.upper=0.06,verbose=3)
tmp <- gl.report.heterozygosity(result,method='ind')
```

`gl.filter.hwe` *Filters loci that show significant departure from Hardy-Weinberg Equilibrium*

Description

This function filters out loci showing significant departure from H-W proportions based on observed frequencies of reference homozygotes, heterozygotes and alternate homozygotes.

Loci are filtered out if they show HWE departure either in any one population (n.pop.threshold =1) or in at least X number of populations (n.pop.threshold > 1).

Usage

```
gl.filter.hwe(
  x,
  subset = "each",
  n.pop.threshold = 1,
  method_sig = "Exact",
  multi_comp = FALSE,
  multi_comp_method = "BY",
  alpha_val = 0.05,
  pvalue_type = "midp",
  cc_val = 0.5,
  min_sample_size = 5,
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object containing the SNP data [required].
`subset`	Way to group individuals to perform H-W tests. Either a vector with population names, 'each', 'all' (see details) [default 'each'].
`n.pop.threshold`	The minimum number of populations where the same locus has to be out of H-W proportions to be removed [default 1].
`method_sig`	Method for determining statistical significance: 'ChiSquare' or 'Exact' [default 'Exact'].
`multi_comp`	Whether to adjust p-values for multiple comparisons [default FALSE].
`multi_comp_method`	Method to adjust p-values for multiple comparisons: 'holm', 'hochberg', 'hommel', 'bonferroni', 'BH', 'BY', 'fdr' (see details) [default 'fdr'].

alpha_val	Level of significance for testing [default 0.05].
pvalue_type	Type of p-value to be used in the Exact method. Either 'dost','selome','midp' (see details) [default 'midp'].
cc_val	The continuity correction applied to the ChiSquare test [default 0.5].
min_sample_size	Minimum number of individuals per population in which perform H-W tests [default 5].
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

There are several factors that can cause deviations from Hardy-Weinberg proportions including: mutation, finite population size, selection, population structure, age structure, assortative mating, sex linkage, nonrandom sampling and genotyping errors. Therefore, testing for Hardy-Weinberg proportions should be a process that involves a careful evaluation of the results, a good place to start is Waples (2015).

Note that tests for H-W proportions are only valid if there is no population substructure (assuming random mating) and have sufficient power only when there is sufficient sample size (n individuals > 15).

Populations can be defined in three ways:

- Merging all populations in the dataset using subset = 'all'.
- Within each population separately using: subset = 'each'.
- Within selected populations using for example: subset = c('pop1','pop2').

Two different statistical methods to test for deviations from Hardy Weinberg proportions:

- The classical chi-square test (method_sig='ChiSquare') based on the function HWChisq of the R package HardyWeinberg. By default a continuity correction is applied (cc_val=0.5). The continuity correction can be turned off (by specifying cc_val=0), for example in cases of extreme allele frequencies in which the continuity correction can lead to excessive type 1 error rates.
- The exact test (method_sig='Exact') based on the exact calculations contained in the function HWExactStats of the R package HardyWeinberg, and described in Wigginton et al. (2005). The exact test is recommended in most cases (Wigginton et al., 2005). Three different methods to estimate p-values (pvalue_type) in the Exact test can be used:
  - 'dost' p-value is computed as twice the tail area of a one-sided test.
  - 'selome' p-value is computed as the sum of the probabilities of all samples less or equally likely as the current sample.
  - 'midp', p-value is computed as half the probability of the current sample + the probabilities of all samples that are more extreme.

  The standard exact p-value is overly conservative, in particular for small minor allele frequencies. The mid p-value ameliorates this problem by bringing the rejection rate closer to the nominal level, at the price of occasionally exceeding the nominal level (Graffelman & Moreno, 2013).

Correction for multiple tests can be applied using the following methods based on the function `p.adjust`:

- 'holm' is also known as the sequential Bonferroni technique (Rice, 1989). This method has a greater statistical power than the standard Bonferroni test, however this method becomes very stringent when many tests are performed and many real deviations from the null hypothesis can go undetected (Waples, 2015).
- 'hochberg' based on Hochberg, 1988.
- 'hommel' based on Hommel, 1988. This method is more powerful than Hochberg's, but the difference is usually small.
- 'bonferroni' in which p-values are multiplied by the number of tests. This method is very stringent and therefore has reduced power to detect multiple departures from the null hypothesis.
- 'BH' based on Benjamini & Hochberg, 1995.
- 'BY' based on Benjamini & Yekutieli, 2001.

The first four methods are designed to give strong control of the family-wise error rate. The last two methods control the false discovery rate (FDR), the expected proportion of false discoveries among the rejected hypotheses. The false discovery rate is a less stringent condition than the family-wise error rate, so these methods are more powerful than the others, especially when number of tests is large. The number of tests on which the adjustment for multiple comparisons is the number of populations times the number of loci.

From v2.1 `gl.filter.hwe` takes the argument `n.pop.threshold`. if `n.pop.threshold > 1` loci will be removed only if they are concurrently significant (after adjustment if applied) out of hwe in `>= n.pop.threshold > 1`.

Value

A genlight object with the loci departing significantly from H-W proportions removed.

Author(s)

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

References

- Benjamini, Y., and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. Annals of Statistics, 29, 1165–1188.
- Graffelman, J. (2015). Exploring Diallelic Genetic Markers: The Hardy Weinberg Package. Journal of Statistical Software 64:1-23.
- Graffelman, J. & Morales-Camarena, J. (2008). Graphical tests for Hardy-Weinberg equilibrium based on the ternary plot. Human Heredity 65:77-84.
- Graffelman, J., & Moreno, V. (2013). The mid p-value in exact tests for Hardy-Weinberg equilibrium. Statistical applications in genetics and molecular biology, 12(4), 433-448.
- Hochberg, Y. (1988). A sharper Bonferroni procedure for multiple tests of significance. Biometrika, 75, 800–803.

- Hommel, G. (1988). A stagewise rejective multiple test procedure based on a modified Bonferroni test. Biometrika, 75, 383–386.
- Rice, W. R. (1989). Analyzing tables of statistical tests. Evolution, 43(1), 223-225.
- Waples, R. S. (2015). Testing for Hardy–Weinberg proportions: have we lost the plot?. Journal of heredity, 106(1), 1-19.
- Wigginton, J.E., Cutler, D.J., & Abecasis, G.R. (2005). A Note on Exact Tests of Hardy-Weinberg Equilibrium. American Journal of Human Genetics 76:887-893.

See Also

gl.report.hwe

Other filter functions: gl.filter.allna(), gl.filter.callrate(), gl.filter.heterozygosity(), gl.filter.ld(), gl.filter.locmetric(), gl.filter.maf(), gl.filter.monomorphs(), gl.filter.overshoot(), gl.filter.parent.offspring(), gl.filter.pa(), gl.filter.rdepth(), gl.filter.reproducibility(), gl.filter.secondaries(), gl.filter.sexlinked(), gl.filter.taglength()

Examples

```
result <- gl.filter.hwe(x = bandicoot.gl)
```

---

gl.filter.ld *Filters loci based on linkage disequilibrium (LD)*

Description

This function uses the statistic set in the parameter stat_keep from function gl.report.ld.map to choose the SNP to keep when two SNPs are in LD. When a SNP is selected to be filtered out in each pairwise comparison, the function stores its name in a list. In subsequent pairwise comparisons, if the SNP is already in the list, the other SNP will be kept.

Usage

```
gl.filter.ld(
  x,
  ld_report,
  threshold = 0.2,
  pop.limit = ceiling(nPop(x)/2),
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
ld_report	Output from function gl.report.ld.map [required].
threshold	Threshold value above which loci will be removed [default 0.2].

pop.limit	Minimum number of populations in which LD should be more than the threshold for a locus to be filtered out. The default value is half of the populations [default ceiling(nPop(x)/2)].
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].

Value

The reduced genlight object.

Author(s)

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

See Also

gl.report.ld.map

Other filter functions: gl.filter.allna(), gl.filter.callrate(), gl.filter.heterozygosity(), gl.filter.hwe(), gl.filter.locmetric(), gl.filter.maf(), gl.filter.monomorphs(), gl.filter.overshoot(), gl.filter.parent.offspring(), gl.filter.pa(), gl.filter.rdepth(), gl.filter.reproducibility(), gl.filter.secondaries(), gl.filter.sexlinked(), gl.filter.taglength()

Examples

```
## Not run:
test <- bandicoot.gl
test <- gl.filter.callrate(test,threshold = 1)
res <- gl.report.ld.map(test)
res_2 <- gl.filter.ld(x=test,ld_report = res)
res_3 <- gl.report.ld.map(res_2)

## End(Not run)
if ((requireNamespace("snpStats", quietly = TRUE)) & (requireNamespace("fields", quietly = TRUE))) {
test <- gl.filter.callrate(platypus.gl, threshold = 1)
test <- gl.filter.monomorphs(test)
test <- test[,1:250]
report <- gl.report.ld.map(test)
res <- gl.filter.ld(x=test,ld_report = report)
}
```

---

gl.filter.locmetric *Filters loci on the basis of numeric information stored in other$loc.metrics in a genlight {adegenet} object*

Description

This script uses any field with numeric values stored in $other$loc.metrics to filter loci. The loci to keep can be within the upper and lower thresholds ('within') or outside of the upper and lower thresholds ('outside').

Usage

```
gl.filter.locmetric(x, metric, upper, lower, keep = "within", verbose = NULL)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
metric	Name of the metric to be used for filtering [required].
upper	Filter upper threshold [required].
lower	Filter lower threshold [required].
keep	Whether keep loci within of upper and lower thresholds or keep loci outside of upper and lower thresholds [within].
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 fields that are included in dartR, and a short description, are found below. Optionally, the user can also set his/her own filter by adding a vector into $other$loc.metrics as shown in the example.

1. SnpPosition - position (zero is position 1) in the sequence tag of the defined SNP variant base.
2. CallRate - proportion of samples for which the genotype call is non-missing (that is, not '-').
3. OneRatioRef - proportion of samples for which the genotype score is 0.
4. OneRatioSnp - proportion of samples for which the genotype score is 2.
5. FreqHomRef - proportion of samples homozygous for the Reference allele.
6. FreqHomSnp - proportion of samples homozygous for the Alternate (SNP) allele.
7. FreqHets - proportion of samples which score as heterozygous, that is, scored as 1.
8. PICRef - polymorphism information content (PIC) for the Reference allele.
9. PICSnp - polymorphism information content (PIC) for the SNP.
10. AvgPIC - average of the polymorphism information content (PIC) of the Reference and SNP alleles.
11. AvgCountRef - sum of the tag read counts for all samples, divided by the number of samples with non-zero tag read counts, for the Reference allele row.
12. AvgCountSnp - sum of the tag read counts for all samples, divided by the number of samples with non-zero tag read counts, for the Alternate (SNP) allele row.
13. RepAvg - proportion of technical replicate assay pairs for which the marker score is consistent.

Value

The reduced genlight dataset.

Author(s)

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

See Also

Other filter functions: gl.filter.allna(), gl.filter.callrate(), gl.filter.heterozygosity(), gl.filter.hwe(), gl.filter.ld(), gl.filter.maf(), gl.filter.monomorphs(), gl.filter.overshoot(), gl.filter.parent.offspring(), gl.filter.pa(), gl.filter.rdepth(), gl.filter.reproducibility(), gl.filter.secondaries(), gl.filter.sexlinked(), gl.filter.taglength()

Examples

```
# adding dummy data
test <- testset.gl
test$other$loc.metrics$test <- 1:nLoc(test)
result <- gl.filter.locmetric(x=test, metric= 'test', upper=255,
lower=200, keep= 'within', verbose=3)
```

---

gl.filter.maf *Filters loci on the basis of minor allele frequency (MAF) in a genlight adegenet object*

Description

This script calculates the minor allele frequency for each locus and updates the locus metadata for FreqHomRef, FreqHomSnp, FreqHets and MAF (if it exists). It then uses the updated metadata for MAF to filter loci.

Usage

```
gl.filter.maf(
  x,
  threshold = 0.01,
  by.pop = FALSE,
  pop.limit = ceiling(nPop(x)/2),
  ind.limit = 10,
  recalc = FALSE,
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors_pop = discrete_palette,
  plot_colors_all = two_colors,
  bins = 25,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
threshold	Threshold MAF – loci with a MAF less than the threshold will be removed. If a value > 1 is provided it will be interpreted as MAC (i.e. the minimum number of times an allele needs to be observed) [default 0.01].

by.pop	Whether MAF should be calculated by population [default FALSE].
pop.limit	Minimum number of populations in which MAF should be less than the threshold for a locus to be filtered out. Only used if by.pop=TRUE. The default value is half of the populations [default ceiling(nPop(x)/2)].
ind.limit	Minimum number of individuals that a population should contain to calculate MAF. Only used if by.pop=TRUE [default 10].
recalc	Recalculate the locus metadata statistics if any individuals are deleted in the filtering [default FALSE].
plot.out	Specify if histograms of call rate, before and after, are to be produced [default TRUE].
plot_theme	User specified theme for the plot [default theme_dartR()].
plot_colors_pop	A color palette for population plots [default discrete_palette].
plot_colors_all	List of two color names for the borders and fill of the overall plot [default two_colors].
bins	Number of bins to display in histograms [default 25].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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

Careful consideration needs to be given to the settings to be used for this fucntion. When the filter is applied globally (i.e. by.pop=FALSE) but the data include multiple population, there is the risk to remove markers because the allele frequencies is low (at global level) but the allele frequencies for the same markers may be high within some of the populations (especially if the per-population sample size is small). Similarly, not always it is a sensible choice to run this function using by.pop=TRUE because allele that are rare in a population may be very common in other, but the (possible) allele frequencies will depend on the sample size within each population. Where the purpose of filtering for MAF is to remove possible spurious alleles (i.e. sequencing errors), it is perhaps better to filter based on the number of times an allele is observed (MAC, Minimum Allele Count), under the assumption that if an allele is observed >MAC, it is fairly rare to be an error. **From v2.1** The threshold can take values > 1. In this case, these are interpreted as a threshold for MAC.

Value

The reduced genlight dataset

Author(s)

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

See Also

Other filter functions: gl.filter.allna(), gl.filter.callrate(), gl.filter.heterozygosity(), gl.filter.hwe(), gl.filter.ld(), gl.filter.locmetric(), gl.filter.monomorphs(), gl.filter.overshoot(), gl.filter.parent.offspring(), gl.filter.pa(), gl.filter.rdepth(), gl.filter.reproducibility(), gl.filter.secondaries(), gl.filter.sexlinked(), gl.filter.taglength()

Examples

```
result <- gl.filter.monomorphs(testset.gl)
result <- gl.filter.maf(result, threshold=0.05, verbose=3)
```

---

gl.filter.monomorphs *Filters monomorphic loci, including those with all NAs*

Description

This script deletes monomorphic loci from a genlight {adegenet} object

A DArT dataset will not have monomorphic loci, but they can arise, along with loci that are scored all NA, when populations or individuals are deleted.

Retaining monomorphic loci unnecessarily increases the size of the dataset and will affect some calculations.

Note that for SNP data, NAs likely represent null alleles; in tag presence/absence data, NAs represent missing values (presence/absence could not be reliably scored)

Usage

```
gl.filter.monomorphs(x, verbose = NULL)
```

Arguments

x	Name of the input genlight object [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].

Value

A genlight object with monomorphic (and all NA) loci removed.

Author(s)

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

See Also

Other filter functions: gl.filter.allna(), gl.filter.callrate(), gl.filter.heterozygosity(), gl.filter.hwe(), gl.filter.ld(), gl.filter.locmetric(), gl.filter.maf(), gl.filter.overshoot(), gl.filter.parent.offspring(), gl.filter.pa(), gl.filter.rdepth(), gl.filter.reproducibility(), gl.filter.secondaries(), gl.filter.sexlinked(), gl.filter.taglength()

Examples

```
# SNP data
  result <- gl.filter.monomorphs(testset.gl, verbose=3)
# Tag P/A data
  result <- gl.filter.monomorphs(testset.gs, verbose=3)
```

`gl.filter.overshoot` *Filters loci for which the SNP has been trimmed from the sequence tag along with the adaptor*

Description

This function checks the position of the SNP within the trimmed sequence tag and identifies those for which the SNP position is outside the trimmed sequence tag. This can happen, rarely, when the sequence containing the SNP resembles the adaptor.

The SNP genotype can still be used in most analyses, but functions like gl2fasta() will present challenges if the SNP has been trimmed from the sequence tag.

Not fatal, but should apply this filter before gl.filter.secondaries, for obvious reasons.

Usage

```
gl.filter.overshoot(x, save2tmp = FALSE, verbose = NULL)
```

Arguments

`x`	Name of the genlight object [required].
`save2tmp`	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
`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].

Value

A new genlight object with the recalcitrant loci deleted

Author(s)

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

See Also

Other filter functions: `gl.filter.allna()`, `gl.filter.callrate()`, `gl.filter.heterozygosity()`, `gl.filter.hwe()`, `gl.filter.ld()`, `gl.filter.locmetric()`, `gl.filter.maf()`, `gl.filter.monomorphs()`, `gl.filter.parent.offspring()`, `gl.filter.pa()`, `gl.filter.rdepth()`, `gl.filter.reproducibility()`, `gl.filter.secondaries()`, `gl.filter.sexlinked()`, `gl.filter.taglength()`

Examples

```
result <- gl.filter.overshoot(testset.gl, verbose=3)
```

`gl.filter.pa` *Filters loci that contain private (and fixed alleles) between two populations*

Description

This script is meant to be used prior to `gl.nhybrids` to maximise the information content of the SNPs used to identify hybrids (currently newhybrids does allow only 200 SNPs). The idea is to use first all loci that have fixed alleles between the potential source populations and then 'fill up' to 200 loci using loci that have private alleles between those. The functions filters for those loci (if invers is set to TRUE, the opposite is returned (all loci that are not fixed and have no private alleles - not sure why yet, but maybe useful.)

Usage

```
gl.filter.pa(x, pop1, pop2, invers = FALSE, verbose = NULL)
```

Arguments

`x`	Name of the genlight object containing the SNP data [required].
`pop1`	Name of the first parental population (in quotes) [required].
`pop2`	Name of the second parental population (in quotes) [required].
`invers`	Switch to filter for all loci that have no private alleles and are not fixed [FALSE].
`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].

Value

The reduced genlight dataset, containing now only fixed and private alleles.

Author(s)

Authors: Bernd Gruber & Ella Kelly (University of Melbourne); Custodian: Luis Mijangos – Post to https://groups.google.com/d/forum/dartr

See Also

Other filter functions: `gl.filter.allna()`, `gl.filter.callrate()`, `gl.filter.heterozygosity()`, `gl.filter.hwe()`, `gl.filter.ld()`, `gl.filter.locmetric()`, `gl.filter.maf()`, `gl.filter.monomorphs()`, `gl.filter.overshoot()`, `gl.filter.parent.offspring()`, `gl.filter.rdepth()`, `gl.filter.reproducibility()`, `gl.filter.secondaries()`, `gl.filter.sexlinked()`, `gl.filter.taglength()`

Examples

```
result <- gl.filter.pa(testset.gl, pop1=pop(testset.gl)[1],
pop2=pop(testset.gl)[2],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.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = two_colors,
  save2tmp = FALSE,
  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.out	Specify if plot is to be produced [default TRUE].
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 two_colors].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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.

Function's output

Plots and table are saved to the temporal directory (tempdir) and can be accessed with the function `gl.print.reports` and listed with the function `gl.list.reports`. Note that they can be accessed only in the current R session because tempdir is cleared each time that the R session is closed.

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.list.reports, gl.report.rdepth , gl.print.reports,gl.report.reproducibility, gl.report.parent.offspri

Other filter functions: gl.filter.allna(), gl.filter.callrate(), gl.filter.heterozygosity(), gl.filter.hwe(), gl.filter.ld(), gl.filter.locmetric(), gl.filter.maf(), gl.filter.monomorphs(), gl.filter.overshoot(), gl.filter.pa(), gl.filter.rdepth(), gl.filter.reproducibility(), gl.filter.secondaries(), gl.filter.sexlinked(), gl.filter.taglength()

Examples

```
out <- gl.filter.parent.offspring(testset.gl[1:10,1:50])
```

---

gl.filter.rdepth *Filters loci based on counts of sequence tags scored at a locus (read depth)*

Description

SNP datasets generated by DArT report AvgCountRef and AvgCountSnp as counts of sequence tags for the reference and alternate alleles respectively. These can be used to back calculate Read Depth. Fragment presence/absence datasets as provided by DArT (SilicoDArT) provide Average Read Depth and Standard Deviation of Read Depth as standard columns in their report.

Filtering on Read Depth using the companion script gl.filter.rdepth can be on the basis of loci with exceptionally low counts, or loci with exceptionally high counts.

Usage

```
gl.filter.rdepth(
  x,
  lower = 5,
  upper = 50,
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = two_colors,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP or tag presence/absence data [required].
lower	Lower threshold value below which loci will be removed [default 5].
upper	Upper threshold value above which loci will be removed [default 50].
plot.out	Specify if plot is to be produced [default TRUE].
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 two_colors].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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

For examples of themes, see:

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

Value

Returns a genlight object retaining loci with a Read Depth in the range specified by the lower and upper threshold.

Author(s)

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

See Also

gl.filter.rdepth

Other filter functions: gl.filter.allna(), gl.filter.callrate(), gl.filter.heterozygosity(), gl.filter.hwe(), gl.filter.ld(), gl.filter.locmetric(), gl.filter.maf(), gl.filter.monomorphs(), gl.filter.overshoot(), gl.filter.parent.offspring(), gl.filter.pa(), gl.filter.reproducibility(), gl.filter.secondaries(), gl.filter.sexlinked(), gl.filter.taglength()

Examples

```
# SNP data
  gl.report.rdepth(testset.gl)
  result <- gl.filter.rdepth(testset.gl, lower=8, upper=50, verbose=3)
# Tag P/A data
  result <- gl.filter.rdepth(testset.gs, lower=8, upper=50, verbose=3)

  res <- gl.filter.rdepth(platypus.gl)
```

`gl.filter.reproducibility` *Filters loci in a genlight {adegenet} object based on average repeatability of alleles at a locus*

Description

SNP datasets generated by DArT have an index, RepAvg, generated by reproducing the data independently for 30 of alleles that give a repeatable result, averaged over both alleles for each locus.

SilicoDArT datasets generated by DArT have a similar index, Reproducibility. For these fragment presence/absence data, repeatability is the percentage of scores that are repeated in the technical replicate dataset.

Usage

```
gl.filter.reproducibility(
  x,
  threshold = 0.99,
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = two_colors,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object containing the SNP data [required].
`threshold`	Threshold value below which loci will be removed [default 0.99].
`plot.out`	If TRUE, displays a plots of the distribution of reproducibility values before and after filtering [default TRUE].
`plot_theme`	Theme for the plot [default theme_dartR()].
`plot_colors`	List of two color names for the borders and fill of the plots [default two_colors].
`save2tmp`	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
`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].

Value

Returns a genlight object retaining loci with repeatability (Repavg or Reproducibility) greater than the specified threshold.

Author(s)

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

See Also

gl.report.reproducibility

Other filter functions: gl.filter.allna(), gl.filter.callrate(), gl.filter.heterozygosity(), gl.filter.hwe(), gl.filter.ld(), gl.filter.locmetric(), gl.filter.maf(), gl.filter.monomorphs(), gl.filter.overshoot(), gl.filter.parent.offspring(), gl.filter.pa(), gl.filter.rdepth(), gl.filter.secondaries(), gl.filter.sexlinked(), gl.filter.taglength()

Examples

```
# SNP data
  gl.report.reproducibility(testset.gl)
  result <- gl.filter.reproducibility(testset.gl, threshold=0.99, verbose=3)
# Tag P/A data
  gl.report.reproducibility(testset.gs)
  result <- gl.filter.reproducibility(testset.gs, threshold=0.99)


  test <- gl.subsample.loci(platypus.gl,n=100)
  res <- gl.filter.reproducibility(test)
```

---

gl.filter.secondaries *Filters loci that represent secondary SNPs in a genlight object*

Description

SNP datasets generated by DArT include fragments with more than one SNP and record them separately with the same CloneID (=AlleleID). These multiple SNP loci within a fragment (secondaries) are likely to be linked, and so you may wish to remove secondaries.

This script filters out all but the first sequence tag with the same CloneID after ordering the genlight object on based on repeatability, avgPIC in that order (method='best') or at random (method='random').

The filter has not been implemented for tag presence/absence data.

Usage

```
gl.filter.secondaries(x, method = "random", verbose = NULL)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
method	Method of selecting SNP locus to retain, 'best' or 'random' [default 'random'].
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].

Value

The genlight object, with the secondary SNP loci removed.

Author(s)

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

See Also

Other filter functions: gl.filter.allna(), gl.filter.callrate(), gl.filter.heterozygosity(), gl.filter.hwe(), gl.filter.ld(), gl.filter.locmetric(), gl.filter.maf(), gl.filter.monomorphs(), gl.filter.overshoot(), gl.filter.parent.offspring(), gl.filter.pa(), gl.filter.rdepth(), gl.filter.reproducibility(), gl.filter.sexlinked(), gl.filter.taglength()

Examples

```
gl.report.secondaries(testset.gl)
result <- gl.filter.secondaries(testset.gl)
```

---

gl.filter.sexlinked *Filters loci that are sex linked*

Description

Alleles unique to the Y or W chromosome and monomorphic on the X chromosomes will appear in the SNP dataset as genotypes that are heterozygotic in all individuals of the heterogametic sex and homozygous in all individuals of the homogametic sex. This function keeps or drops loci with alleles that behave in this way, as putative sex specific SNP markers.

Usage

```
gl.filter.sexlinked(
  x,
  sex = NULL,
  filter = NULL,
  read.depth = 0,
  t.het = 0.1,
  t.hom = 0.1,
  t.pres = 0.1,
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = three_colors,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP or presence/absence (SilicoDArT) data [required].
sex	Factor that defines the sex of individuals. See explanation in details [default NULL].
filter	Either 'keep' to keep sex linked markers only or 'drop' to drop sex linked markers [required].
read.depth	Additional filter option to keep only loci above a certain read.depth. Default to 0, which means read.depth is not taken into account [default 0].
t.het	Tolerance in the heterogametic sex, that is t.het=0.05 means that 5% of the heterogametic sex can be homozygous and still be regarded as consistent with a sex specific marker [default 0.1].
t.hom	Tolerance in the homogametic sex, that is t.hom=0.05 means that 5% of the homogametic sex can be heterozygous and still be regarded as consistent with a sex specific marker [default 0.1].
t.pres	Tolerance in presence, that is t.pres=0.05 means that a silicodart marker can be present in either of the sexes and still be regarded as a sex-linked marker [default 0.1].
plot.out	Creates a plot that shows the heterozygosity of males and females at each loci be regarded as consistent with a sex specific marker [default TRUE].
plot_theme	Theme for the plot. See Details for options [default theme_dartR()].
plot_colors	List of three color names for the not sex-linked loci, for the sex-linked loci and for the area in which sex-linked loci appear [default three_colors].
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

Sex of the individuals for which sex is known with certainty can be provided via a factor (equal to the length of the number of individuals) or to be held in the variable `x@other$ind.metrics$sex`. Coding is: M for male, F for female, U or NA for unknown/missing. The script abbreviates the entries here to the first character. So, coding of 'Female' and 'Male' works as well. Character are also converted to upper cases.

' Function's output

This function creates also a plot that shows the heterozygosity of males and females at each loci for SNP data or percentage of present/absent in the case of SilicoDArT data.

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 (filter = 'keep': sex linked loci, filter='drop', everything except sex linked loci).

Author(s)

Arthur Georges, Bernd Gruber & Floriaan Devloo-Delva (Post to https://groups.google.com/d/forum/dartr)

See Also

Other filter functions: gl.filter.allna(), gl.filter.callrate(), gl.filter.heterozygosity(), gl.filter.hwe(), gl.filter.ld(), gl.filter.locmetric(), gl.filter.maf(), gl.filter.monomorphs(), gl.filter.overshoot(), gl.filter.parent.offspring(), gl.filter.pa(), gl.filter.rdepth(), gl.filter.reproducibility(), gl.filter.secondaries(), gl.filter.taglength()

Examples

```
out <- gl.filter.sexlinked(testset.gl, filter='drop')

out <- gl.filter.sexlinked(testset.gs, filter='drop')
```

---

gl.filter.taglength *Filters loci in a genlight {adegenet} object based on sequence tag length*

Description

SNP datasets generated by DArT typically have sequence tag lengths ranging from 20 to 69 base pairs.

Usage

```
gl.filter.taglength(x, lower = 20, upper = 69, verbose = NULL)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
lower	Lower threshold value below which loci will be removed [default 20].
upper	Upper threshold value above which loci will be removed [default 69].
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].

Value

Returns a genlight object retaining loci with a sequence tag length in the range specified by the lower and upper threshold.

Author(s)

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

See Also

Other filter functions: gl.filter.allna(), gl.filter.callrate(), gl.filter.heterozygosity(), gl.filter.hwe(), gl.filter.ld(), gl.filter.locmetric(), gl.filter.maf(), gl.filter.monomorphs(), gl.filter.overshoot(), gl.filter.parent.offspring(), gl.filter.pa(), gl.filter.rdepth(), gl.filter.reproducibility(), gl.filter.secondaries(), gl.filter.sexlinked()

Examples

```
# SNP data
  gl.report.taglength(testset.gl)
  result <- gl.filter.taglength(testset.gl,lower=60)
  gl.report.taglength(result)
# Tag P/A data
  gl.report.taglength(testset.gs)
  result <- gl.filter.taglength(testset.gs,lower=60)
  gl.report.taglength(result)

  test <- gl.subsample.loci(platypus.gl, n =100)
  res <- gl.report.taglength(test)
```

---

gl.fixed.diff *Generates a matrix of fixed differences and associated statistics for populations taken pairwise*

Description

This script takes SNP data or sequence tag P/A data grouped into populations in a genlight object (DArTSeq) and generates a matrix of fixed differences between populations taken pairwise

Usage

```
gl.fixed.diff(
  x,
  tloc = 0,
  test = FALSE,
  delta = 0.02,
  alpha = 0.05,
  reps = 1000,
  mono.rm = TRUE,
  pb = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing SNP genotypes or tag P/A data (SilicoDArT) or an object of class 'fd' [required].
tloc	Threshold defining a fixed difference (e.g. 0.05 implies 95:5 vs 5:95 is fixed) [default 0].
test	If TRUE, calculate p values for the observed fixed differences [default FALSE].
delta	Threshold value for the true population minor allele frequency (MAF) from which resultant sample fixed differences are considered true positives [default 0.02].
alpha	Level of significance used to display non-significant differences between populations as they are compared pairwise [default 0.05].
reps	Number of replications to undertake in the simulation to estimate probability of false positives [default 1000].
mono.rm	If TRUE, loci that are monomorphic across all individuals are removed before beginning computations [default TRUE].
pb	If TRUE, show a progress bar on time consuming loops [default FALSE].
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

A fixed difference at a locus occurs when two populations share no alleles or where all members of one population has a sequence tag scored, and all members of the other population has the sequence tag absent. The challenge with this approach is that when sample sizes are finite, fixed differences will occur through sampling error, compounded when many loci are examined. Simulations suggest that sample sizes of n1=5 and n2=5 are adequate to reduce the probability of [experiment-wide] type 1 error to negligible levels [ploidy=2]. A warning is issued if comparison between two populations involves sample sizes less than 5, taking into account allele drop-out.

Optionally, if test=TRUE, the script will test the fixed differences between final OTUs for statistical significance, using simulation, and then further amalgamate populations that for which there are no significant fixed differences at a specified level of significance (alpha). To avoid conflation of true fixed differences with false positives in the simulations, it is necessary to decide a threshold value (delta) for extreme true allele frequencies that will be considered fixed for practical purposes. That is, fixed differences in the sample set will be considered to be positives (not false positives) if they arise from true allele frequencies of less than 1-delta in one or both populations. The parameter delta is typically set to be small (e.g. delta = 0.02).

NOTE: The above test will only be calculated if tloc=0, that is, for analyses of absolute fixed differences. The test applies in comparisons of allopatric populations only. For sympatric populations, use gl.pval.sympatry().

An absolute fixed difference is as defined above. However, one might wish to score fixed differences at some lower level of allele frequency difference, say where percent allele frequencies are 95,5 and 5,95 rather than 100:0 and 0:100. This adjustment can be done with the tloc parameter. For example, tloc=0.05 means that SNP allele frequencies of 95,5 and 5,95 percent will be regarded as fixed when comparing two populations at a locus.

Value

A list of Class 'fd' containing the gl object and square matrices, as follows:

1. $gl – the output genlight object;
2. $fd – raw fixed differences;
3. $pcfd – percent fixed differences;
4. $nobs – mean no. of individuals used in each comparison;
5. $nloc – total number of loci used in each comparison;
6. $expfpos – if test=TRUE, the expected count of false positives for each comparison [by simulation];
7. $sdfpos – if test=TRUE, the standard deviation of the count of false positives for each comparison [by simulation];
8. $prob – if test=TRUE, the significance of the count of fixed differences [by simulation])

Author(s)

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

See Also

is.fixed

Examples

```
fd <- gl.fixed.diff(testset.gl, tloc=0, verbose=3 )
fd <- gl.fixed.diff(testset.gl, tloc=0, test=TRUE, delta=0.02, reps=100, verbose=3 )
```

---

gl.fst.pop *Calculates a pairwise Fst values for populations in a genlight object*

Description

This script calculates pairwise Fst values based on the implementation in the StAMPP package (?stamppFst). It allows to run bootstrap to estimate probability of Fst values to be different from zero. For detailed information please check the help pages (?stamppFst).

Usage

```
gl.fst.pop(x, nboots = 100, percent = 95, nclusters = 1, verbose = NULL)
```

Arguments

x	Name of the genlight containing the SNP genotypes [required].
nboots	Number of bootstraps to perform across loci to generate confidence intervals and p-values [default 100].
percent	Percentile to calculate the confidence interval around [default 95].
nclusters	Number of processor threads or cores to use during calculations [default 1].
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].

Value

A matrix of distances between populations (class dist), if nboots =1, otherwise a list with Fsts (in a matrix), Pvalues (a matrix of pvalues), Bootstraps results (data frame of all runs). Hint: Use `as.matrix(as.dist(fsts))` if you want to have a squared matrix with symmetric entries returned, instead of a dist object.

Author(s)

Bernd Gruber (bugs? Post to https://groups.google.com/d/forum/dartr)

Examples

```
test <- gl.filter.callrate(platypus.gl,threshold = 1)
test <- gl.filter.monomorphs(test)
out <- gl.fst.pop(test, nboots=1)
```

---

gl.genleastcost *Performs least-cost path analysis based on a friction matrix*

Description

This function calculates the pairwise distances (Euclidean, cost path distances and genetic distances) of populations using a friction matrix and a spatial genind object. The genind object needs to have coordinates in the same projected coordinate system as the friction matrix. The friction matrix can be either a single raster of a stack of several layers. If a stack is provided the specified cost distance is calculated for each layer in the stack. The output of this function can be used with the functions wassermann or lgrMMRR to test for the significance of a layer on the genetic structure.

Usage

```
gl.genleastcost(
  x,
  fric.raster,
  gen.distance,
  NN = NULL,
  pathtype = "leastcost",
```

```
  plotpath = TRUE,
  theta = 1,
  verbose = NULL
)
```

Arguments

x	A spatial genind object. See ?popgenreport how to provide coordinates in genind objects [required].
fric.raster	A friction matrix [required].
gen.distance	Specification which genetic distance method should be used to calculate pairwise genetic distances between populations ('D', 'Gst.Nei', 'Gst.Hedrick') or individuals ('Smouse', 'Kosman', 'propShared') [required].
NN	Number of neighbours used when calculating the cost distance (possible values 4, 8 or 16). As the default is NULL a value has to be provided if pathtype='leastcost'. NN=8 is most commonly used. Be aware that linear structures may cause artefacts in the least-cost paths, therefore inspect the actual least-cost paths in the provided output [default NULL].
pathtype	Type of cost distance to be calculated (based on function in the gdistance package. Available distances are 'leastcost', 'commute' or 'rSPDistance'. See functions in the gdistance package for futher explanations. If the path type is set to 'leastcost' then paths and also pathlength are returned [default 'leastcost'].
plotpath	switch if least cost paths should be plotted (works only if pathtype='leastcost'. Be aware this slows down the computation, but it is recommended to do this to check least cost paths visually.
theta	value needed for rSPDistance function. See rSPDistance in package gdistance [default 1].
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].

Value

Returns a list that consists of four pairwise distance matrices (Euclidean, Cost, length of path and genetic) and the actual paths as spatial line objects.

Author(s)

Bernd Gruber (bugs? Post to https://groups.google.com/d/forum/dartr)

References

- Cushman, S., Wasserman, T., Landguth, E. and Shirk, A. (2013). Re-Evaluating Causal Modeling with Mantel Tests in Landscape Genetics. Diversity, 5(1), 51-72.
- Landguth, E. L., Cushman, S. A., Schwartz, M. K., McKelvey, K. S., Murphy, M. and Luikart, G. (2010). Quantifying the lag time to detect barriers in landscape genetics. Molecular ecology, 4179-4191.

- Wasserman, T. N., Cushman, S. A., Schwartz, M. K. and Wallin, D. O. (2010). Spatial scaling and multi-model inference in landscape genetics: Martes americana in northern Idaho. Landscape Ecology, 25(10), 1601-1612.

See Also

landgenreport, popgenreport, wassermann, lgrMMRR

Examples

```
## Not run:
data(possums.gl)
library(raster)  #needed for that example
landscape.sim <- readRDS(system.file('extdata','landscape.sim.rdata',
package='dartR'))
glc <- gl.genleastcost(x=possums.gl,fric.raster=landscape.sim ,
gen.distance = 'D', NN=8, pathtype = 'leastcost',plotpath = TRUE)
library(PopGenReport)
PopGenReport::wassermann(eucl.mat = glc$eucl.mat, cost.mat = glc$cost.mats,
 gen.mat = glc$gen.mat)
lgrMMRR(gen.mat = glc$gen.mat, cost.mats = glc$cost.mats,
eucl.mat = glc$eucl.mat)

## End(Not run)
```

---

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 = discrete_palette,
  palette_convergent = convergent_palette,
  legendx = 0,
  legendy = 0.5,
  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	A discrete palette for the color of populations or a list with as many colors as there are populations in the dataset [default discrete_palette].
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].
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",
  link.color = NULL,
  link.size = 2,
  relatedness_factor = 0.125,
  title = "Network based on a genomic relationship matrix",
  palette_discrete = NULL,
  save2tmp = FALSE,
  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'].
`link.color`	Color palette for links [default NULL].
`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 palette for the color of populations or a list with as many colors as there are populations in the dataset [default NULL].

`save2tmp`
: If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].

`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 takes a genomic relationship matrix (GRM) generated by the gl.grm function to represent the relationship among individuals in the dataset as a network diagram. To generate the GRM, the function gl.grm uses the function A.mat from package rrBLUP, which implements the approach developed by Endelman and Jannink (2012).

The GRM is an estimate of the proportion of alleles that two individuals have in common. It is generated by estimating the covariance of the genotypes between two individuals, i.e. how much genotypes in the two individuals correspond with each other. This covariance depends on the probability that alleles at a random locus are identical by state (IBS). Two alleles are IBS if they represent the same allele. Two alleles are identical by descent (IBD) if one is a physical copy of the other or if they are both physical copies of the same ancestral allele. Note that IBD is complicated to determine. IBD implies IBS, but not conversely. However, as the number of SNPs in a dataset increases, the mean probability of IBS approaches the mean probability of IBD.

It follows that the off-diagonal elements of the GRM are two times the kinship coefficient, i.e. the probability that two alleles at a random locus drawn from two individuals are IBD. Additionally, the diagonal elements of the GRM are 1+f, where f is the inbreeding coefficient of each individual, i.e. the probability that the two alleles at a random locus are IBD.

Choosing a meaningful threshold to represent the relationship between individuals is tricky because IBD is not an absolute state but is relative to a reference population for which there is generally little information so that we can estimate the kinship of a pair of individuals only relative to some other quantity. To deal with this, we can use the average inbreeding coefficient of the diagonal elements as the reference value. For this, the function subtracts 1 from the mean of the diagonal elements of the GRM. In a second step, the off-diagonal elements are divided by 2, and finally, the mean of the diagonal elements is subtracted from each off-diagonal element after dividing them by 2. This approach is similar to the one used by Goudet et al. (2018).

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.loci(t1,n = 100)
# relatedness matrix
res <- gl.grm(t1,plotheatmap = FALSE)
# relatedness network
res2 <- gl.grm.network(res,t1,relatedness_factor = 0.125)
}
```

gl.He *Estimates expected Heterozygosity*

Description

Estimates expected Heterozygosity

Usage

```
gl.He(gl)
```

Arguments

gl	A genlight object [required]

Value

A simple vector whit Ho for each loci

Author(s)

Bernd Gruber (Post to https://groups.google.com/d/forum/dartr)

gl.Ho *Estimates observed Heterozygosity*

Description

Estimates observed Heterozygosity

Usage

```
gl.Ho(gl)
```

Arguments

gl	A genlight object [required]

Value

A simple vector whit Ho for each loci

Author(s)

Bernd Gruber (bugs? Post to https://groups.google.com/d/forum/dartr)

gl.hwe.pop *Performs Hardy-Weinberg tests over loci and populations*

Description

Hardy-Weinberg tests are performed for each loci in each of the populations as defined by the pop slot in a genlight object.

Usage

```
gl.hwe.pop(
  x,
  alpha_val = 0.05,
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = c("gray90", "deeppink"),
  HWformat = FALSE,
  verbose = NULL
)
```

Arguments

x	A genlight object with a population defined [pop(x) does not return NULL].
alpha_val	Level of significance for testing [default 0.05].
plot.out	If TRUE, returns a plot object compatible with ggplot, otherwise returns a dataframe [default TRUE].
plot_theme	User specified theme [default theme_dartR()].
plot_colors	Vector with two color names for the borders and fill [default two_colors]. [default discrete_palette].
HWformat	Switch if data should be returned in HWformat (counts of Genotypes to be used in package HardyWeinberg)
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

This function employs the HardyWeinberg package, which needs to be installed. The function that is used is HWExactStats, but there are several other great functions implemented in the package regarding HWE. Therefore, this function can return the data in the format expected by the HWE package expects, via HWformat=TRUE and then use this to run other functions of the package.

This functions performs a HWE test for every population (rows) and loci (columns) and returns a true false matrix. True is reported if the p-value of an HWE-test for a particular loci and population was below the specified threshold (alpha_val, default=0.05). The thinking behind this approach is that loci that are not in HWE in several populations have most likely to be treated (e.g. filtered if loci under selection are of interest). If plot=TRUE a barplot on the loci and the sum of deviation

over all population is returned. Loci that deviate in the majority of populations can be identified via colSums on the resulting matrix.

Plot themes can be obtained from

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

Resultant ggplots and the tabulation are saved to the session's temporary directory.

Value

The function returns a list with up to three components:

- 'HWE' is the matrix over loci and populations
- 'plot' is a plot (ggplot) which shows the significant results for population and loci (can be amended further using ggplot syntax)
- 'HWEformat=TRUE' the 'HWformat' entails SNP data for each population in 'HardyWeinberg'-format to be used with other functions of the package (e.g HWPerm or HWExactPrevious).

Author(s)

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

Examples

```
out <- gl.hwe.pop(bandicoot.gl[,1:33], alpha_val=0.05, plot.out=TRUE, HWformat=FALSE)
```

---

gl.ibd *Performs isolation by distance analysis*

Description

This function performs an isolation by distance analysis based on a Mantel test and also produces an isolation by distance plot. If a genlight object with coordinates is provided, then an Euclidean and genetic distance matrices are calculated. #'

Usage

```
gl.ibd(
  x = NULL,
  distance = "Fst",
  coordinates = "latlon",
  Dgen = NULL,
  Dgeo = NULL,
  Dgeo_trans = "Dgeo",
  Dgen_trans = "Dgen",
  permutations = 999,
```

```
  plot.out = TRUE,
  paircols = NULL,
  plot_theme = theme_dartR(),
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Genlight object. If provided a standard analysis on Fst/1-Fst and log(distance) is performed [required].
distance	Type of distance that is calculated and used for the analysis. Can be either population based 'Fst' [stamppFst], 'D' [stamppNeisD] or individual based 'propShared', [gl.propShared], 'euclidean' [gl.dist.ind, method='Euclidean'] [default "Fst"].
coordinates	Can be either 'latlon', 'xy' or a two column data.frame with column names 'lat','lon', 'x', 'y'). Coordinates are provided via `gl@other$latlon` ['latlon'] or via `gl@other$xy` ['xy']. If latlon data will be projected to meters using Mercator system [google maps] or if xy then distance is directly calculated on the coordinates.
Dgen	Genetic distance matrix if no genlight object is provided [default NULL].
Dgeo	Euclidean distance matrix if no genlight object is provided [default NULL].
Dgeo_trans	Transformation to be used on the Euclidean distances. See Dgen_trans [default "Dgeo"].
Dgen_trans	You can provide a formula to transform the genetic distance. The transformation can be applied as a formula using Dgen as the variable to be transformed. For example: `Dgen_trans = 'Dgen/(1-Dgen)'`. Any valid R expression can be used here [default 'Dgen', which is the identity function.]
permutations	Number of permutations in the Mantel test [default 999].
plot.out	Should an isolation by distance plot be returned [default TRUE].
paircols	Should pairwise dots colored by 'pop'ulation/'ind'ividual pairs [default 'pop']. You can color pairwise individuals by pairwise population colors.
plot_theme	Theme for the plot. See details for options [default theme_dartR()].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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

Currently pairwise Fst and D between populations and 1-propShared and Euclidean distance between individuals are implemented. Coordinates are expected as lat long and converted to Google Earth Mercator projection. If coordinates are already projected, provide them at the x@other$xy slot.

You can provide also your own genetic and Euclidean distance matrices. The function is based on the code provided by the adegenet tutorial (http://adegenet.r-forge.r-project.org/files/tutorial-basics.pdf), using the functions mantel (package vegan), stamppFst, stamppNeisD (package StAMPP) and gl.propShared or gl.dist.ind. For transformation you need to have the dismo package installed. As a new feature you can plot pairwise relationship using double colored points (paircols=TRUE). Pairwise relationship can be visualised via populations or individuals, depending which distance is calculated. Please note: Often a problem arises, if an individual based distance is calculated (e.g. propShared) and some individuals have identical coordinates as this results in distances of zero between those pairs of individuals. If the standard transformation [log(Dgeo)] is used, this results in an infinite value, because of trying to calculate'log(0)'. To avoid this, the easiest fix is to change the transformation from log(Dgeo) to log(Dgeo+1) or you could add some "noise" to the coordinates of the individuals (e.g. +- 1m, but be aware if you use lat lon then you rather want to add +0.00001 degrees or so).

Value

Returns a list of the following components: Dgen (the genetic distance matrix), Dgeo (the Euclidean distance matrix), Mantel (the statistics of the Mantel test).

Author(s)

Bernd Gruber (bugs? Post to https://groups.google.com/d/forum/dartr)

References

Rousset, F. (1997). Genetic differentiation and estimation of gene flow from F-statistics under isolation by distance. Genetics, 145(4), 1219-1228.

See Also

mantel, stamppFst

Examples

```
#because of speed only the first 100 loci
ibd <- gl.ibd(bandicoot.gl[,1:100], Dgeo_trans='log(Dgeo)' ,Dgen_trans='Dgen/(1-Dgen)')
#because of speed only the first 10 individuals)
ibd <- gl.ibd(bandicoot.gl[1:10,], distance='euclidean', paircols='pop', Dgeo_trans='Dgeo')


#only first 100 loci
ibd <- gl.ibd(bandicoot.gl[,1:100])
```

`gl.impute` *Imputates missing data*

Description

This function imputes genotypes on a population-by-population basis, where populations can be considered panmictic, or imputes the state for presence-absence data.

Usage

```
gl.impute(
  x,
  method = "neighbour",
  fill.residual = TRUE,
  parallel = FALSE,
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object containing the SNP or presence-absence data [required].
`method`	Imputation method, either "frequency" or "HW" or "neighbour" or "random" [default "HW"].
`fill.residual`	Should any residual missing values remaining after imputation be set to 0, 1, 2 at random, taking into account global allele frequencies at the particular locus [default TRUE].
`parallel`	A logical indicating whether multiple cores -if available- should be used for the computations (TRUE), or not (FALSE); requires the package parallel to be installed [default FALSE].
`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

We recommend that imputation be performed on sampling locations, before any aggregation. Imputation is achieved by replacing missing values using either of two methods:

- If "frequency", genotypes scored as missing at a locus in an individual are imputed using the average allele frequencies at that locus in the population from which the individual was drawn.
- If "HW", genotypes scored as missing at a locus in an individual are imputed by sampling at random assuming Hardy-Weinberg equilibrium. Applies only to genotype data.
- If "neighbour", substitute the missing values for the focal individual with the values taken from the nearest neighbour. Repeat with next nearest and so on until all missing values are replaced.

- if "random", missing data are substituted by random values (0, 1 or 2).

The nearest neighbour is the one with the smallest Euclidean distance in all the dataset.

The advantage of this approach is that it works regardless of how many individuals are in the population to which the focal individual belongs, and the displacement of the individual is haphazard as opposed to:

(a) Drawing the individual toward the population centroid (HW and Frequency).

(b) Drawing the individual toward the global centroid (glPCA).

Note that loci that are missing for all individuals in a population are not imputed with method 'frequency' or 'HW'. Consider using the function gl.filter.allna with by.pop=TRUE to remove them first.

Value

A genlight object with the missing data imputed.

Author(s)

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

Examples

```
require("dartR.data")
# SNP genotype data
gl <- gl.filter.callrate(platypus.gl,threshold=0.95)
gl <- gl.filter.allna(gl)
gl <- gl.impute(gl,method="neighbour")
# Sequence Tag presence-absence data
gs <- gl.filter.callrate(testset.gs,threshold=0.95)
gl <- gl.filter.allna(gl)
gs <- gl.impute(gs, method="neighbour")

gs <- gl.impute(platypus.gl,method ="random")
```

---

gl.install.vanilla.dartR

Installs all required packages for using all functions available in dartR

Description

The function compares the installed packages with the the currently available ones on CRAN. Be aware this function only works if a version of dartR is already installed on your system. You can choose if you also want to have a specific version of dartR installed ('CRAN', 'master', 'beta' or 'dev'). 'master' , 'beta' and 'dev' are installed from Github. Be aware that the dev version from github is not fully tested and most certainly will contain untested functions.

Usage

```
gl.install.vanilla.dartR(flavour = NULL, verbose = NULL)
```

Arguments

flavour	The version of R you want to install. If NULL then only packages needed for the current version will be installed. If 'CRAN' current CRAN version will be installed. 'master' installs the GitHub master branch, 'beta' installs the latest stable version, and 'dev' installs the experimental development branch from GitHub [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

Returns a message if the installation was successful/required.

Author(s)

Custodian: Bernd Gruber (Post to https://groups.google.com/d/forum/dartr)

---

gl.join *Combines two genlight objects*

Description

This function combines two genlight objects and their associated metadata. The history associated with the two genlight objects is cleared from the new genlight object. The individuals/samples must be the same in each genlight object.

The function is typically used to combine datasets from the same service where the files have been split because of size limitations. The data is read in from multiple csv files, then the resultant genlight objects are combined.

Usage

```
gl.join(x1, x2, verbose = NULL)
```

Arguments

x1	Name of the first genlight object [required].
x2	Name of the first genlight object [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].

Value

A new genlight object

Author(s)

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

Examples

```
x1 <- testset.gl[,1:100]
x1@other$loc.metrics <-  testset.gl@other$loc.metrics[1:100,]
nLoc(x1)
x2 <- testset.gl[,101:150]
x2@other$loc.metrics <-  testset.gl@other$loc.metrics[101:150,]
nLoc(x2)
gl <- gl.join(x1, x2, verbose=2)
nLoc(gl)
```

---

gl.keep.ind — *Removes all but the specified individuals from a genlight {adegenet} object*

Description

The script, having deleted individuals, optionally identifies resultant monomorphic loci or loci with all values missing and deletes them (using gl.filter.monomorphs.r). The script also optionally recalculates statistics made redundant by the deletion of individuals from the dataset.

The script returns a genlight object with the individuals deleted and, optionally, the recalculated locus metadata.

Usage

```
gl.keep.ind(x, ind.list, recalc = FALSE, mono.rm = FALSE, verbose = NULL)
```

Arguments

x	Name of the genlight object containing the SNP or presence/absence (SilicoDArT) data [required].
ind.list	A list of individuals to be removed [required].
recalc	Recalculate the locus metadata statistics [default FALSE].
mono.rm	Remove monomorphic loci [default FALSE].
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 with the reduced data

Author(s)

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

See Also

gl.drop.ind to drop rather than keep specified individuals

Examples

```
# SNP data
  gl2 <- gl.keep.ind(testset.gl, ind.list=c('AA019073','AA004859'))
# Tag P/A data
 gs2 <- gl.keep.ind(testset.gs, ind.list=c('AA020656','AA19077','AA004859'))
```

---

gl.keep.loc *Removes all but the specified loci from a genlight {adegenet} object*

Description

The script returns a genlight object with the all but the specified loci deleted.

Usage

```
gl.keep.loc(x, loc.list = NULL, first = NULL, last = NULL, verbose = NULL)
```

Arguments

x	Name of the genlight object containing SNP genotypes or presence/absence data [required].
loc.list	A list of loci to be kept [required, if loc.range not specified].
first	First of a range of loci to be kept [required, if loc.list not specified].
last	Last of a range of loci to be kept [if not specified, last locus in the dataset].
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 with the reduced data

Author(s)

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

See Also

gl.drop.loc to drop rather than keep specified loci

Examples

```
# SNP data
  gl2 <- gl.keep.loc(testset.gl, loc.list=c('100051468|42-A/T', '100049816-51-A/G'))
# Tag P/A data
  gs2 <- gl.keep.loc(testset.gs, loc.list=c('20134188','19249144'))
```

`gl.keep.pop` *Removes all but the specified populations from a genlight object*

Description

Individuals are assigned to populations based on the specimen metadata data file (csv) used with gl.read.dart().

The script, having deleted the specified populations, optionally identifies resultant monomorphic loci or loci with all values missing and deletes them (using gl.filter.monomorphs.r). The script also optionally recalculates statistics made redundant by the deletion of individuals from the dataset.

The script returns a genlight object with the new population assignments and the recalculated locus metadata.

Usage

```
gl.keep.pop(
  x,
  pop.list,
  as.pop = NULL,
  recalc = FALSE,
  mono.rm = FALSE,
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object containing the SNP or presence/absence (Silico-DArT) data [required].
`pop.list`	A list of populations to be kept [required].
`as.pop`	Assign another metric to represent population [default NULL].
`recalc`	Recalculate the locus metadata statistics [default FALSE].
`mono.rm`	Remove monomorphic loci [default FALSE].
`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 with the reduced data

Author(s)

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

See Also

gl.drop.pop to drop rather than keep specified populations

Examples

```
# SNP data
  gl2 <- gl.keep.pop(testset.gl, pop.list=c('EmsubRopeMata', 'EmvicVictJasp'))
  gl2 <- gl.keep.pop(testset.gl, pop.list=c('EmsubRopeMata', 'EmvicVictJasp'),
  mono.rm=TRUE,recalc=TRUE)
  gl2 <- gl.keep.pop(testset.gl, pop.list=c('Female'),as.pop='sex')
# Tag P/A data
  gs2 <- gl.keep.pop(testset.gs, pop.list=c('EmsubRopeMata','EmvicVictJasp'))
```

---

gl.ld.distance — *Plots linkage disequilibrium against distance by population disequilibrium patterns*

Description

The function creates a plot showing the pairwise LD measure against distance in number of base pairs pooled over all the chromosomes and a red line representing the threshold (R.squared = 0.2) that is commonly used to imply that two loci are unlinked (Delourme et al., 2013; Li et al., 2014).

Usage

```
gl.ld.distance(
  ld_report,
  ld_resolution = 1e+05,
  pop_colors = NULL,
  plot_theme = NULL,
  plot.out = TRUE,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

ld_report	Output from function gl.report.ld.map [required].
ld_resolution	Resolution at which LD should be reported in number of base pairs [default NULL].
pop_colors	A color palette for box plots by population or a list with as many colors as there are populations in the dataset [default NULL].

plot_theme	User specified theme [default NULL].
plot.out	Specify if plot is to be produced [default TRUE].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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].

Value

A dataframe with information of LD against distance by population.

Author(s)

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

References

- Delourme, R., Falentin, C., Fomeju, B. F., Boillot, M., Lassalle, G., André, I., . . . Marty, A. (2013). High-density SNP-based genetic map development and linkage disequilibrium assessment in Brassica napusL. BMC genomics, 14(1), 120.
- Li, X., Han, Y., Wei, Y., Acharya, A., Farmer, A. D., Ho, J., . . . Brummer, E. C. (2014). Development of an alfalfa SNP array and its use to evaluate patterns of population structure and linkage disequilibrium. PLoS One, 9(1), e84329.

See Also

Other ld functions: gl.ld.haplotype()

Examples

```
if ((requireNamespace("snpStats", quietly = TRUE)) & (requireNamespace("fields", quietly = TRUE))) {
require("dartR.data")
x <- platypus.gl
x <- gl.filter.callrate(x,threshold = 1)
x <- gl.filter.monomorphs(x)
x$position <- x$other$loc.metrics$ChromPos_Platypus_Chrom_NCBIv1
x$chromosome <- as.factor(x$other$loc.metrics$Chrom_Platypus_Chrom_NCBIv1)
ld_res <- gl.report.ld.map(x,ld_max_pairwise = 10000000)
ld_res_2 <- gl.ld.distance(ld_res,ld_resolution= 1000000)
}
```

`gl.ld.haplotype` *Visualize patterns of linkage disequilibrium and identification of haplotypes*

Description

This function plots a Linkage disequilibrium (LD) heatmap, where the colour shading indicates the strength of LD. Chromosome positions (Mbp) are shown on the horizontal axis, and haplotypes appear as triangles and delimited by dark yellow vertical lines. Numbers identifying each haplotype are shown in the upper part of the plot.

The heatmap also shows heterozygosity for each SNP.

The function identifies haplotypes based on contiguous SNPs that are in linkage disequilibrium using as threshold `ld_threshold_haplo` and containing more than `min_snps` SNPs.

Usage

```
gl.ld.haplotype(
  x,
  pop_name = NULL,
  chrom_name = NULL,
  ld_max_pairwise = 1e+07,
  maf = 0.05,
  ld_stat = "R.squared",
  ind.limit = 10,
  min_snps = 10,
  ld_threshold_haplo = 0.5,
  coordinates = NULL,
  color_haplo = "viridis",
  color_het = "deeppink",
  plot.out = TRUE,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object containing the SNP data [required].
`pop_name`	Name of the population to analyse. If NULL all the populations are analised [default NULL].
`chrom_name`	Nme of the chromosome to analyse. If NULL all the chromosomes are analised [default NULL].
`ld_max_pairwise`	Maximum distance in number of base pairs at which LD should be calculated [default 10000000].

maf	Minor allele frequency (by population) threshold to filter out loci. If a value > 1 is provided it will be interpreted as MAC (i.e. the minimum number of times an allele needs to be observed) [default 0.05].
ld_stat	The LD measure to be calculated: "LLR", "OR", "Q", "Covar", "D.prime", "R.squared", and "R". See ld (package snpStats) for details [default "R.squared"].
ind.limit	Minimum number of individuals that a population should contain to take it in account to report loci in LD [default 10].
min_snps	Minimum number of SNPs that should have a haplotype to call it [default 10].
ld_threshold_haplo	Minimum LD between adjacent SNPs to call a haplotype [default 0.5].
coordinates	A vector of two elements with the start and end coordinates in base pairs to which restrict the analysis e.g. c(1,1000000) [default NULL].
color_haplo	Color palette for haplotype plot. See details [default "viridis"].
color_het	Color for heterozygosity [default "deeppink"].
plot.out	Specify if heatmap plot is to be produced [default TRUE].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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 information for SNP's position should be stored in the genlight accessor "@position" and the SNP's chromosome name in the accessor "@chromosome" (see examples). The function will then calculate LD within each chromosome.

The output of the function includes a table with the haplotypes that were identified and their location.

Colors of the heatmap (color_haplo) are based on the function scale_fill_viridis from package viridis. Other color palettes options are "magma", "inferno", "plasma", "viridis", "cividis", "rocket", "mako" and "turbo".

Value

A table with the haplotypes that were identified.

Author(s)

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

See Also

Other ld functions: gl.ld.distance()

Examples

```
require("dartR.data")
x <- platypus.gl
x <- gl.filter.callrate(x,threshold = 1)
x <- gl.keep.pop(x, pop.list = "TENTERFIELD")
x$chromosome <- as.factor(x$other$loc.metrics$Chrom_Platypus_Chrom_NCBIv1)
x$position <- x$other$loc.metrics$ChromPos_Platypus_Chrom_NCBIv1
ld_res <- gl.ld.haplotype(x,chrom_name = "NC_041728.1_chromosome_1",
                          ld_max_pairwise = 10000000 )
```

---

gl.LDNe *Estimates effective population size using the Linkage Disequilibrium method based on NeEstimator (V2)*

Description

This function is basically a convenience function that runs the LD Ne estimator using Neestimator2 (http://www.molecularfisherieslaboratory.com.au/neestimator-software/) within R using the provided genlight object. To be able to do so, the software has to be downloaded from their website and the appropriate executable Ne2-1 has to be copied into the path as specified in the function (see example below).

Usage

```
gl.LDNe(
  x,
  outfile = "genepopLD.txt",
  outpath = tempdir(),
  neest.path = getwd(),
  critical = 0,
  singleton.rm = TRUE,
  mating = "random",
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors_pop = discrete_palette,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
outfile	File name of the output file with all results from Neestimator 2 [default 'genepopLD.txt'].
outpath	Path where to save the output file. Use outpath=getwd() or outpath='.' when calling this function to direct output files to your working directory [default tempdir(), mandated by CRAN].

neest.path	Path to the folder of the NE2-1 file. Please note there are 3 different executables depending on your OS: Ne2-1.exe (=Windows), Ne2-1M (=Mac), Ne2-1L (=Linux). You only need to point to the folder (the function will recognise which OS you are running) [default getwd()].
critical	(vector of) Critical values that are used to remove alleles based on their minor allele frequency. This can be done before using the gl.filter.maf function, therefore the default is set to 0 (no loci are removed). To run for MAF 0 and MAF 0.05 at the same time specify: critical = c(0,0.05) [default 0].
singleton.rm	Whether to remove singleton alleles [default TRUE].
mating	Formula for Random mating='random' or monogamy= 'monogamy' [default 'random'].
plot.out	Specify if plot is to be produced [default TRUE].
plot_theme	User specified theme [default theme_dartR()].
plot_colors_pop	A discrete palette for population colors or a list with as many colors as there are populations in the dataset [default discrete_palette].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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].

Value

Dataframe with the results as table

Author(s)

Custodian: Bernd Gruber (Post to https://groups.google.com/d/forum/dartr)

Examples

```
## Not run:
# SNP data (use two populations and only the first 100 SNPs)
pops <- possums.gl[1:60,1:100]
nes <- gl.LDNe(pops, outfile="popsLD.txt", outpath=tempdir(),
neest.path = "./path_to Ne-21",
critical=c(0,0.05), singleton.rm=TRUE, mating='random')
nes

## End(Not run)
```

gl.list.reports *Prints dartR reports saved in tempdir*

Description

Prints dartR reports saved in tempdir

Usage

```
gl.list.reports()
```

Value

Prints a table with all reports saved in tempdir. Currently the style cannot be changed.

Author(s)

Bernd Gruber & Luis Mijangos (bugs? Post to https://groups.google.com/d/forum/dartr)

See Also

gl.print.reports

Examples

```
## Not run:
gl.report.callrate(testset.gl,save2tmp=TRUE)
gl.list.reports()

## End(Not run)
```

gl.load *Loads an object from compressed binary format produced by gl.save()*

Description

This is a wrapper for readRDS()

Usage

```
gl.load(file, verbose = NULL)
```

Arguments

file	Name of the file to receive the binary version of the object [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

The script loads the object from the current workspace and returns the gl object.

Value

The loaded object

Author(s)

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

See Also

gl.save

Examples

```
gl.save(testset.gl,file.path(tempdir(),'testset.rds'))
gl <- gl.load(file.path(tempdir(),'testset.rds'))
```

---

gl.make.recode.ind *Creates a proforma recode_ind file for reassigning individual (=specimen) names*

Description

Renaming individuals may be required when there have been errors in labelling arising in the process from sample to DArT files. There may be occasions where renaming individuals is required for preparation of figures. Caution needs to be exercised because of the potential for breaking the 'chain of evidence' between the samples themselves and the analyses. Recoding individuals can be done with a recode table (csv).

Usage

```
gl.make.recode.ind(
  x,
  out.recode.file = "default_recode_ind.csv",
  outpath = tempdir(),
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
out.recode.file	File name of the output file (including extension) [default default_recode_ind.csv].

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

Details

This script facilitates the construction of a recode table by producing a proforma file with current individual (=specimen) names in two identical columns. Edit the second column to reassign individual names. Use keyword Delete to delete an individual.

Apply the recoding using gl.recode.ind(). Deleting individuals can potentially generate monomorphic loci or loci with all values missing. Clean this up with gl.filter.monomorphic().

Value

A vector containing the new individual names.

Author(s)

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

Examples

```
result <- gl.make.recode.ind(testset.gl, out.recode.file ='Emmac_recode_ind.csv',outpath=tempdir())
```

---

gl.make.recode.pop *Creates a proforma recode_pop_table file for reassigning population names*

Description

Renaming populations may be required when there have been errors in assignment arising in the process from sample to DArT files or when one wishes to amalgamate populations, or delete populations. Recoding populations can also be done with a recode table (csv).

Usage

```
gl.make.recode.pop(
  x,
  out.recode.file = "recode_pop_table.csv",
  outpath = tempdir(),
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
out.recode.file	File name of the output file (including extension) [default recode_pop_table.csv].
outpath	Path where to save the output file [default tempdir(), mandated by CRAN]. Use outpath=getwd() or outpath='.' when calling this function to direct output files to your 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].

Details

This script facilitates the construction of a recode table by producing a proforma file with current population names in two identical columns. Edit the second column to reassign populations. Use keyword Delete to delete a population.

Apply the recoding using gl.recode.pop().

Value

A vector containing the new population names.

Author(s)

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

Examples

```
result <- gl.make.recode.pop(testset.gl,out.recode.file='test.csv',outpath=tempdir(),verbose=2)
```

---

gl.map.interactive *Creates an interactive map (based on latlon) from a genlight object*

Description

Creates an interactive map (based on latlon) from a genlight object

Usage

```
gl.map.interactive(
  x,
  matrix = NULL,
  standard = TRUE,
  symmetric = TRUE,
  pop.labels = TRUE,
  pop.labels.cex = 12,
  ind.circles = TRUE,
```

```
  ind.circle.cols = NULL,
  ind.circle.cex = 10,
  ind.circle.transparency = 0.8,
  palette_links = NULL,
  leg_title = NULL,
  provider = "Esri.NatGeoWorldMap",
  verbose = NULL
)
```

Arguments

x	A genlight object (including coordinates within the latlon slot) [required].
matrix	A distance matrix between populations or individuals. The matrix is visualised as lines between individuals/populations. If matrix is asymmetric two lines with arrows are plotted [default NULL].
standard	If a matrix is provided line width will be standardised to be between 1 to 10, if set to true, otherwise taken as given [default TRUE].
symmetric	If a symmetric matrix is provided only one line is drawn based on the lower triangle of the matrix. If set to false arrows indicating the direction are used instead [default TRUE].
pop.labels	Population labels at the center of the individuals of populations [default TRUE].
pop.labels.cex	Size of population labels [default 12].
ind.circles	Should individuals plotted as circles [default TRUE].
ind.circle.cols	Colors of circles. Colors can be provided as usual by names (e.g. "black") and are re-cycled. So a color c("blue","red") colors individuals alternatively between blue and red using the genlight object order of individuals. For transparency see parameter ind.circle.transparency. Defaults to rainbow colors by population if not provided. If you want to have your own colors for each population, check the platypus.gl example below.
ind.circle.cex	(size or circles in pixels) [default 10].
ind.circle.transparency	Transparency of circles between 0=invisible and 1=no transparency. Defaults to 0.8.
palette_links	Color palette for the links in case a matrix is provided [default NULL].
leg_title	Legend's title for the links in case a matrix is provided [default NULL].
provider	Passed to leaflet [default "Esri.NatGeoWorldMap"].
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

A wrapper around the **leaflet** package. For possible background maps check as specified via the provider: http://leaflet-extras.github.io/leaflet-providers/preview/index.html

The palette_links argument can be any of the following: A character vector of RGB or named colors. Examples: palette(), c("#000000", "#0000FF", "#FFFFFF"), topo.colors(10)

The name of an RColorBrewer palette, e.g. "BuPu" or "Greens".

The full name of a viridis palette: "viridis", "magma", "inferno", or "plasma".

A function that receives a single value between 0 and 1 and returns a color. Examples: colorRamp(c("#000000", "#FFFFFF"), interpolate = "spline").

Value

plots a map

Author(s)

Bernd Gruber – Post to https://groups.google.com/d/forum/dartr

Examples

```
require("dartR.data")
gl.map.interactive(bandicoot.gl)
cols <- c("red","blue","yellow")[as.numeric(pop(platypus.gl))]
gl.map.interactive(platypus.gl, ind.circle.cols=cols, ind.circle.cex=10,
ind.circle.transparency=0.5)
```

---

gl.map.structure *Maps a STRUCTURE plot using a genlight object*

Description

This function takes the output of plotstructure (the q matrix) and maps the q-matrix across using the population centers from the genlight object that was used to run the structure analysis via gl.run.structure) and plots the typical structure bar plots on a spatial map, providing a barplot for each subpopulation. Therefore it requires coordinates from a genlight object. This kind of plots should support the interpretation of the spatial structure of a population, but in principle is not different from gl.plot.structure

Usage

```
gl.map.structure(
  qmat,
  x,
  K,
  provider = "Esri.NatGeoWorldMap",
  scalex = 1,
  scaley = 1,
  movepops = NULL,
  pop.labels = TRUE,
  pop.labels.cex = 12
)
```

Arguments

qmat	Q-matrix from a structure run followed by a clumpp run object [from gl.run.structure and gl.plot.structure] [required].
x	Name of the genlight object containing the coordinates in the \@other$latlon slot to calculate the population centers [required].
K	The number for K to be plotted [required].
provider	Provider passed to leaflet. Check providers for a list of possible backgrounds [default "Esri.NatGeoWorldMap"].
scalex	Scaling factor to determine the size of the bars in x direction [default 1].
scaley	Scaling factor to determine the size of the bars in y direction [default 1].
movepops	A two-dimensional data frame that allows to move the center of the barplots manually in case they overlap. Often if populations are horizontally close to each other. This needs to be a data.frame of the dimensions [rows=number of populations, columns = 2 (lon/lat)]. For each population you have to specify the x and y (lon and lat) units you want to move the center of the plot, (see example for details) [default NULL].
pop.labels	Switch for population labels below the parplots [default TRUE].
pop.labels.cex	Size of population labels [default 12].

Details

Creates a mapped version of structure plots. For possible background maps check as specified via the provider: http://leaflet-extras.github.io/leaflet-providers/preview/index.html. You may need to adjust scalex and scaley values [default 1], as the size depends on the scale of the map and the position of the populations.

Value

An interactive map that shows the structure plots broken down by population.

returns the map and a list of the qmat split into sorted matrices per population. This can be used to create your own map.

Author(s)

Bernd Gruber (Post to https://groups.google.com/d/forum/dartr)

References

- Pritchard, J.K., Stephens, M., Donnelly, P. (2000) Inference of population structure using multilocus genotype data. Genetics 155, 945-959.
- Archer, F. I., Adams, P. E. and Schneiders, B. B. (2016) strataG: An R package for manipulating, summarizing and analysing population genetic data. Mol Ecol Resour. doi:10.1111/1755-0998.12559
- Evanno, G., Regnaut, S., and J. Goudet. 2005. Detecting the number of clusters of individuals using the software STRUCTURE: a simulation study. Molecular Ecology 14:2611-2620.

- Mattias Jakobsson and Noah A. Rosenberg. 2007. CLUMPP: a cluster matching and permutation program for dealing with label switching and multimodality in analysis of population structure. Bioinformatics 23(14):1801-1806. Available at clumpp

See Also

gl.run.structure, clumpp, gl.plot.structure

Examples

```
## Not run:
#bc <- bandicoot.gl[,1:100]
#sr <- gl.run.structure(bc, k.range = 2:5, num.k.rep = 3, exec = './structure.exe')
#ev <- gl.evanno(sr)
#ev
#qmat <- gl.plot.structure(sr, k=2:4)#' #head(qmat)
#gl.map.structure(qmat, bc,K=3)
#gl.map.structure(qmat, bc,K=4)
#move population 4 (out of 5) 0.5 degrees to the right and populations 1
#0.3 degree to the north of the map.
#mp <- data.frame(lon=c(0,0,0,0.5,0), lat=c(-0.3,0,0,0,0))
#gl.map.structure(qmat, bc,K=4, movepops=mp)

## End(Not run)
```

---

gl.merge.pop *Merges two or more populations in a genlight object into one population*

Description

Individuals are assigned to populations based on the specimen metadata data file (csv) used with gl.read.dart().

This script assigns individuals from two nominated populations into a new single population. It can also be used to rename populations.

The script returns a genlight object with the new population assignments.

Usage

```
gl.merge.pop(x, old = NULL, new = NULL, verbose = NULL)
```

Arguments

x	Name of the genlight object containing SNP genotypes [required].
old	A list of populations to be merged [required].
new	Name of the new 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 or as specified using gl.set.verbosity].

Value

A genlight object with the new population assignments.

Author(s)

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

Examples

```
gl <- gl.merge.pop(testset.gl, old=c('EmsubRopeMata','EmvicVictJasp'), new='Outgroup')
```

---

gl.nhybrids — *Creates an input file for the program NewHybrids and runs it if NewHybrids is installed*

Description

This function compares two sets of parental populations to identify loci that exhibit a fixed difference, returns an genlight object with the reduced data, and creates an input file for the program NewHybrids using the top 200 (or hard specified loc.limit) loci. In the absence of two identified parental populations, the script will select a random set 200 loci only (method='random') or the first 200 loci ranked on information content (method='AvgPIC').

A fixed difference occurs when a SNP allele is present in all individuals of one population and absent in the other. There is provision for setting a level of tolerance, e.g. threshold = 0.05 which considers alleles present at greater than 95 a fixed difference. Only the 200 loci are retained, because of limitations of NewHybids.

If you specify a directory for the NewHybrids executable file, then the script will create the input file from the SNP data then run NewHybrids. If the directory is set to NULL, the execution will stop once the input file (default='nhyb.txt') has been written to disk. Note: the executable option will not work on a Mac; Mac users should generate the NewHybrids input file and run this on their local installation of NewHybrids.

Refer to the New Hybrids manual for further information on the parameters to set – http://ib.berkeley.edu/labs/slatkin/eriq/soft

It is important to stringently filter the data on RepAvg and CallRate if using the random option. One might elect to repeat the analysis (method='random') and combine the resultant posterior probabilities should 200 loci be considered insufficient.

The F1 individuals should be homozygous at all loci for which the parental populations are fixed and different, assuming parental populations have been specified. Sampling errors can result in this not being the case, especially where the sample sizes for the parental populations are small. Alternatively, the threshold for posterior probabilities used to determine assignment (pprob) or the definition of a fixed difference (threshold) may be too lax. To assess the error rate in the determination of assignment of F1 individuals, a plot of the frequency of homozygous reference, heterozygotes and homozygous alternate (SNP) can be produced by setting plot=TRUE (the default).

Usage

```
gl.nhybrids(
  gl,
  outpath = tempdir(),
  p0 = NULL,
  p1 = NULL,
  threshold = 0,
  method = "random",
  plot = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = two_colors,
  pprob = 0.95,
  nhyb.directory = NULL,
  BurnIn = 10000,
  sweeps = 10000,
  GtypFile = "TwoGensGtypFreq.txt",
  AFPriorFile = NULL,
  PiPrior = "Jeffreys",
  ThetaPrior = "Jeffreys",
  verbose = NULL
)
```

Arguments

gl	Name of the genlight object containing the SNP data [required].
outpath	Path where to save the output file [default tempdir()].
p0	List of populations to be regarded as parental population 0 [default NULL].
p1	List of populations to be regarded as parental population 1 [default NULL].
threshold	Sets the level at which a gene frequency difference is considered to be fixed [default 0].
method	Specifies the method (random or AvgPIC) to select 200 loci for NewHybrids [default random].
plot	If TRUE, a plot of the frequency of homozygous reference, heterozygotes and homozygous alternate (SNP) is produced for the F1 individuals [default TRUE, applies only if both parental populations are specified].
plot_theme	User specified theme [default theme_dartR()].
plot_colors	Vector with two color names for the borders and fill [default two_colors].
pprob	Threshold level for assignment to likelihood bins [default 0.95, used only if plot=TRUE].
nhyb.directory	Directory that holds the NewHybrids executable file e.g. C:/NewHybsPC [default NULL].
BurnIn	Number of sweeps to use in the burn in [default 10000].
sweeps	Number of sweeps to use in computing the actual Monte Carlo averages [default 10000].

GtypFile	Name of a file containing the genotype frequency classes [default TwoGensGtypFreq.txt].
AFPriorFile	Name of the file containing prior allele frequency information [default NULL].
PiPrior	Jeffreys-like priors or Uniform priors for the parameter pi [default Jeffreys].
ThetaPrior	Jeffreys-like priors or Uniform priors for the parameter theta [default Jeffreys].
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

The reduced genlight object, if parentals are provided; output of NewHybrids is saved to the working directory.

Author(s)

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

References

Anderson, E.C. and Thompson, E.A.(2002). A model-based method for identifying species hybrids using multilocus genetic data. Genetics. 160:1217-1229.

Examples

```
## Not run:
m <- gl.nhybrids(testset.gl,
p0=NULL, p1=NULL,
nhyb.directory='D:/workspace/R/NewHybsPC', # Specify as necessary
outpath="D:/workspace",  # Specify as necessary, usually getwd() [= workspace]
BurnIn=100,
sweeps=100,
verbose=3)

## End(Not run)
```

---

gl.outflank *Identifies loci under selection per population using the outflank method of Whitlock and Lotterhos (2015)*

Description

Identifies loci under selection per population using the outflank method of Whitlock and Lotterhos (2015)

Usage

```
gl.outflank(
  gi,
  plot = TRUE,
  LeftTrimFraction = 0.05,
  RightTrimFraction = 0.05,
  Hmin = 0.1,
  qthreshold = 0.05,
  ...
)
```

Arguments

gi	A genlight or genind object, with a defined population structure [required].
plot	A switch if a barplot is wanted [default TRUE].
LeftTrimFraction	The proportion of loci that are trimmed from the lower end of the range of Fst before the likelihood function is applied [default 0.05].
RightTrimFraction	The proportion of loci that are trimmed from the upper end of the range of Fst before the likelihood function is applied [default 0.05].
Hmin	The minimum heterozygosity required before including calculations from a locus [default 0.1].
qthreshold	The desired false discovery rate threshold for calculating q-values [default 0.05].
...	additional parameters (see documentation of outflank on github).

Details

This function is a wrapper around the outflank function provided by Whitlock and Lotterhos. To be able to run this function the packages qvalue (from bioconductor) and outflank (from github) needs to be installed. To do so see example below.

Value

Returns an index of outliers and the full outflank list

References

Whitlock, M.C. and Lotterhos K.J. (2015) Reliable detection of loci responsible for local adaptation: inference of a neutral model through trimming the distribution of Fst. The American Naturalist 186: 24 - 36.

Github repository: Whitlock & Lotterhos: https://github.com/whitlock/OutFLANK (Check the readme.pdf within the repository for an explanation. Be aware you now can run OufFLANK from a genlight object)

See Also

utils.outflank, utils.outflank.plotter, utils.outflank.MakeDiploidFSTMat

Examples

```
gl.outflank(bandicoot.gl, plot = TRUE)
```

`gl.pcoa` *Ordination applied to genotypes in a genlight object (PCA), in an fd object, or to a distance matrix (PCoA)*

Description

This function takes the genotypes for individuals and undertakes a Pearson Principal Component analysis (PCA) on SNP or Tag P/A (SilicoDArT) data; it undertakes a Gower Principal Coordinate analysis (PCoA) if supplied with a distance matrix. Technically, any distance matrix can be represented in an ordinated space using PCoA.

Usage

```
gl.pcoa(
  x,
  nfactors = 5,
  correction = NULL,
  mono.rm = TRUE,
  parallel = FALSE,
  n.cores = 16,
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = two_colors,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object or fd object containing the SNP data, or a distance matrix of type dist [required].
`nfactors`	Number of axes to retain in the output of factor scores [default 5].
`correction`	Method applied to correct for negative eigenvalues, either 'lingoes' or 'cailliez' [Default NULL].
`mono.rm`	If TRUE, remove monomorphic loci [default TRUE].
`parallel`	TRUE if parallel processing is required (does fail under Windows) [default FALSE].
`n.cores`	Number of cores to use if parallel processing is requested [default 16].
`plot.out`	If TRUE, a diagnostic plot is displayed showing a scree plot for the "informative" axes and a histogram of eigenvalues of the remaining "noise" axes [Default TRUE].

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 plot [default two_colors].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
verbose	verbose= 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 function is essentially a wrapper for glPca adegenet or pcoa {ape} with default settings apart from those specified as parameters in this function. **Sources of stress in the visual representation**

While, technically, any distance matrix can be represented in an ordinated space, the representation will not typically be exact.There are three major sources of stress in a reduced-representation of distances or dissimilarities among entities using PCA or PCoA. By far the greatest source comes from the decision to select only the top two or three axes from the ordinated set of axes derived from the PCA or PCoA. The representation of the entities such a heavily reduced space will not faithfully represent the distances in the input distance matrix simply because of the loss of information in deeper informative dimensions. For this reason, it is not sensible to be too precious about managing the other two sources of stress in the visual representation.

The measure of distance between entities in a PCA is the Pearson Correlation Coefficient, essentially a standardized Euclidean distance. This is both a metric distance and a Euclidean distance. In PCoA, the second source of stress is the choice of distance measure or dissimilarity measure. While any distance or dissimilarity matrix can be represented in an ordinated space, the distances between entities can be faithfully represented in that space (that is, without stress) only if the distances are metric. Furthermore, for distances between entities to be faithfully represented in a rigid Cartesian space, the distance measure needs to be Euclidean. If this is not the case, the distances between the entities in the ordinated visualized space will not exactly represent the distances in the input matrix (stress will be non-zero). This source of stress will be evident as negative eigenvalues in the deeper dimensions.

A third source of stress arises from having a sparse dataset, one with missing values. This affects both PCA and PCoA. If the original data matrix is not fully populated, that is, if there are missing values, then even a Euclidean distance matrix will not necessarily be 'positive definite'. It follows that some of the eigenvalues may be negative, even though the distance metric is Euclidean. This issue is exacerbated when the number of loci greatly exceeds the number of individuals, as is typically the case when working with SNP data. The impact of missing values can be minimized by stringently filtering on Call Rate, albeit with loss of data. An alternative is given in a paper 'Honey, I shrunk the sample covariance matrix' and more recently by Ledoit and Wolf (2018), but their approach has not been implemented here.

The good news is that, unless the sum of the negative eigenvalues, arising from a non-Euclidean distance measure or from missing values, approaches those of the final PCA or PCoA axes to be displayed, the distortion is probably of no practical consequence and certainly not comparable to the stress arising from selecting only two or three final dimensions out of several informative dimensions for the visual representation.

Function's output

Two diagnostic plots are produced. The first is a Scree Plot, showing the percentage variation explained by each of the PCA or PCoA axes, for those axes that explain more than the original

variables (loci) on average. That is, only informative axes are displayed. The scree plot informs the number of dimensions to be retained in the visual summaries. As a rule of thumb, axes with more than 10

The second graph shows the distribution of eigenvalues for the remaining uninformative (noise) axes, including those with negative eigenvalues.

Action is recommended (verbose >= 2) if the negative eigenvalues are dominant, their sum approaching in magnitude the eigenvalues for axes selected for the final visual solution.

Output is a glPca object conforming to adegenet::glPca but with only the following retained.

- $call - The call that generated the PCA/PCoA
- $eig - Eigenvalues – All eigenvalues (positive, null, negative).
- $scores - Scores (coefficients) for each individual
- $loadings - Loadings of each SNP for each principal component

Plots and table were saved to the temporal directory (tempdir) and can be accessed with the function `gl.print.reports` and listed with the function `gl.list.reports`. Note that they can be accessed only in the current R session because tempdir is cleared each time that the R session is closed.

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/

PCA was developed by Pearson (1901) and Hotelling (1933), whilst the best modern reference is Jolliffe (2002). PCoA was developed by Gower (1966) while the best modern reference is Legendre & Legendre (1998).

Value

An object of class pcoa containing the eigenvalues and factor scores

Author(s)

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

References

- Cailliez, F. (1983) The analytical solution of the additive constant problem. Psychometrika, 48, 305-308.
- Gower, J. C. (1966) Some distance properties of latent root and vector methods used in multivariate analysis. Biometrika, 53, 325-338.
- Hotelling, H., 1933. Analysis of a complex of statistical variables into Principal Components. Journal of Educational Psychology 24:417-441, 498-520.
- Jolliffe, I. (2002) Principal Component Analysis. 2nd Edition, Springer, New York.
- Ledoit, O. and Wolf, M. (2018). Analytical nonlinear shrinkage of large-dimensional covariance matrices. University of Zurich, Department of Economics, Working Paper No. 264, Revised version. Available at SSRN: https://ssrn.com/abstract=3047302 or http://dx.doi.org/10.2139/ssrn.3047302

- Legendre, P. and Legendre, L. (1998). Numerical Ecology, Volume 24, 2nd Edition. Elsevier Science, NY.
- Lingoes, J. C. (1971) Some boundary conditions for a monotone analysis of symmetric matrices. Psychometrika, 36, 195-203.
- Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. Philosophical Magazine. Series 6, vol. 2, no. 11, pp. 559-572.

See Also

gl.pcoa.plot

Examples

```
## Not run:
gl <- possums.gl
# PCA (using SNP genlight object)
pca <- gl.pcoa(possums.gl[1:50,],verbose=2)
gl.pcoa.plot(pca,gl)

gs <- testset.gs
levels(pop(gs))<-c(rep('Coast',5),rep('Cooper',3),rep('Coast',5),
rep('MDB',8),rep('Coast',6),'Em.subglobosa','Em.victoriae')

# PCA (using SilicoDArT genlight object)
pca <- gl.pcoa(gs)
gl.pcoa.plot(pca,gs)

# Collapsing pops to OTUs using Fixed Difference Analysis (using fd object)
fd <- gl.fixed.diff(testset.gl)
fd <- gl.collapse(fd)
pca <- gl.pcoa(fd)
gl.pcoa.plot(pca,fd$gl)

# Using a distance matrix
D <- gl.dist.ind(testset.gs, method='jaccard')
pcoa <- gl.pcoa(D,correction="cailliez")
gl.pcoa.plot(pcoa,gs)

## End(Not run)
```

---

gl.pcoa.plot *Bivariate or trivariate plot of the results of an ordination generated using gl.pcoa()*

Description

This script takes output from the ordination generated by gl.pcoa() and plots the individuals classified by population.

Usage

```
gl.pcoa.plot(
  glPca,
  x,
  scale = FALSE,
  ellipse = FALSE,
  plevel = 0.95,
  pop.labels = "pop",
  interactive = FALSE,
  as.pop = NULL,
  hadjust = 1.5,
  vadjust = 1,
  xaxis = 1,
  yaxis = 2,
  zaxis = NULL,
  pt.size = 2,
  pt.colors = NULL,
  pt.shapes = NULL,
  label.size = 1,
  axis.label.size = 1.5,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

glPca	Name of the PCA or PCoA object containing the factor scores and eigenvalues [required].
x	Name of the genlight object or fd object containing the SNP genotypes or Tag P/A (SilicoDArT) genotypes or the Distance Matrix used to generate the ordination [required].
scale	If TRUE, scale the x and y axes in proportion to % variation explained [default FALSE].
ellipse	If TRUE, display ellipses to encapsulate points for each population [default FALSE].
plevel	Value of the percentile for the ellipse to encapsulate points for each population [default 0.95].
pop.labels	How labels will be added to the plot ['none'|'pop'|'legend', default = 'pop'].
interactive	If TRUE then the populations are plotted without labels, mouse-over to identify points [default FALSE].
as.pop	Assign another metric to represent populations for the plot [default NULL].
hadjust	Horizontal adjustment of label position in 2D plots [default 1.5].
vadjust	Vertical adjustment of label position in 2D plots [default 1].
xaxis	Identify the x axis from those available in the ordination (xaxis <= nfactors) [default 1].

yaxis	Identify the y axis from those available in the ordination (yaxis <= nfactors) [default 2].
zaxis	Identify the z axis from those available in the ordination for a 3D plot (zaxis <= nfactors) [default NULL].
pt.size	Specify the size of the displayed points [default 2].
pt.colors	Optionally provide a vector of nPop colors (run gl.select.colors() for color options) [default NULL].
pt.shapes	Optionally provide a vector of nPop shapes (run gl.select.shapes() for shape options) [default NULL].
label.size	Specify the size of the point labels [default 1].
axis.label.size	Specify the size of the displayed axis labels [default 1.5].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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 factor scores are taken from the output of gl.pcoa() and the population assignments are taken from from the original data file. In the bivariate plots, the specimens are shown optionally with adjacent labels and enclosing ellipses. Population labels on the plot are shuffled so as not to overlap (using package {directlabels}). This can be a bit clunky, as the labels may be some distance from the points to which they refer, but it provides the opportunity for moving labels around using graphics software (e.g. Adobe Illustrator).

3D plotting is activated by specifying a zaxis.

Any pair or trio of axes can be specified from the ordination, provided they are within the range of the nfactors value provided to gl.pcoa(). In the 2D plots, axes can be scaled to represent the proportion of variation explained. In any case, the proportion of variation explained by each axis is provided in the axis label.

Colors and shapes of the points can be altered by passing a vector of shapes and/or a vector of colors. These vectors can be created with gl.select.shapes() and gl.select.colors() and passed to this script using the pt.shapes and pt.colors parameters.

Points displayed in the ordination can be identified if the option interactive=TRUE is chosen, in which case the resultant plot is ggplotly() friendly. Identification of points is by moving the mouse over them. Refer to the plotly package for further information. The interactive option is automatically enabled for 3D plotting.

Author(s)

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

See Also

gl.pcoa

Other Exploration/visualisation functions: gl.select.colors(), gl.select.shapes(), gl.smearplot()

Examples

```
# SET UP DATASET
gl <- testset.gl
levels(pop(gl))<-c(rep('Coast',5),rep('Cooper',3),rep('Coast',5),
rep('MDB',8),rep('Coast',7),'Em.subglobosa','Em.victoriae')
# RUN PCA
pca<-gl.pcoa(gl,nfactors=5)
# VARIOUS EXAMPLES
gl.pcoa.plot(pca, gl, ellipse=TRUE, plevel=0.95, pop.labels='pop',
axis.label.size=1, hadjust=1.5,vadjust=1)
gl.pcoa.plot(pca, gl, ellipse=TRUE, plevel=0.99, pop.labels='legend',
axis.label.size=1)
gl.pcoa.plot(pca, gl, ellipse=TRUE, plevel=0.99, pop.labels='legend',
axis.label.size=1.5,scale=TRUE)
gl.pcoa.plot(pca, gl, ellipse=TRUE, axis.label.size=1.2, xaxis=1, yaxis=3,
scale=TRUE)
gl.pcoa.plot(pca, gl, pop.labels='none',scale=TRUE)
gl.pcoa.plot(pca, gl, axis.label.size=1.2, interactive=TRUE)
gl.pcoa.plot(pca, gl, ellipse=TRUE, plevel=0.99, xaxis=1, yaxis=2, zaxis=3)
# color AND SHAPE ADJUSTMENTS
shp <- gl.select.shapes(select=c(16,17,17,0,2))
col <- gl.select.colors(library='brewer',palette='Spectral',ncolors=11,
select=c(1,9,3,11,11))
gl.pcoa.plot(pca, gl, ellipse=TRUE, plevel=0.95, pop.labels='pop',
pt.colors=col, pt.shapes=shp, axis.label.size=1, hadjust=1.5,vadjust=1)
gl.pcoa.plot(pca, gl, ellipse=TRUE, plevel=0.99, pop.labels='legend',
 pt.colors=col, pt.shapes=shp, axis.label.size=1)

 test <- gl.pcoa(platypus.gl)
 gl.pcoa.plot(glPca = test, x = platypus.gl)
```

---

gl.percent.freq *Generates percentage allele frequencies by locus and population*

Description

This is a support script, to take SNP data or SilicoDArT presence/absence data grouped into populations in a genlight object {adegenet} and generate a table of allele frequencies for each population and locus

Usage

```
gl.percent.freq(x, verbose = NULL)
```

Arguments

x	Name of the genlight object containing the SNP or Tag P/A (SilicoDArT) data [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].

Value

A matrix with allele (SNP data) or presence/absence frequencies (Tag P/A data) broken down by population and locus

Author(s)

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

Examples

```
m <-  gl.percent.freq(testset.gl)
```

---

gl.play.history *Replays the history and applies it to a genlight object*

Description

Replays the history and applies it to a genlight object

Usage

```
gl.play.history(x, history = NULL, verbose = 0)
```

Arguments

x	A genlight object (with a history slot) [optional].
history	If no history is provided the complete history of x is used (recreating the identical object x). If history is a vector it indicates which which part of the history of x is used [c(1,3,4) uses the first, third and forth entry from x@other$history]. Or a simple link to a history slot of another genlight object (e.g. codex2@other$history[c(1,4,5)]). [optional].
verbose	If set to one then history commands are printed, which may facilitate reading the output [default 0].

Details

This function basically allows to create a 'template history' (=set of filters) and apply them to any other genlight object. Histories can also be saved and loaded (see. gl.save.history and gl.load.history).

Value

Returns a genlight object that was created by replaying the provided applied to the genlight object x. Please note you can 'mix' histories or part of them and apply them to different genlight objects. If the history does not contain gl.read.dart, histories of x and history are concatenated.

Author(s)

Bernd Gruber (bugs? Post to https://groups.google.com/d/forum/dartr).

Examples

```
## Not run:
dartfile <- system.file('extdata','testset_SNPs_2Row.csv', package='dartR')
metadata <- system.file('extdata','testset_metadata.csv', package='dartR')
gl <- gl.read.dart(dartfile, ind.metafile = metadata, probar=FALSE)
gl2 <- gl.filter.callrate(gl, method='loc', threshold=0.9)
gl3 <- gl.filter.callrate(gl2, method='ind', threshold=0.95)
#Now 'replay' part of the history 'onto' another genlight object
#bc.fil <- gl.play.history(gl.compliance.check(bandicoot.gl),
#history=gl3@other$history[c(2,3)], verbose=1)
#gl.print.history(bc.fil)

## End(Not run)
```

---

gl.plot.heatmap *Represents a distance matrix as a heatmap*

Description

The script plots a heat map to represent the distances in the distance or dissimilarity matrix. This function is a wrapper for heatmap.2 (package gplots).

Usage

```
gl.plot.heatmap(D, palette_divergent = diverging_palette, verbose = NULL, ...)
```

Arguments

D	Name of the distance matrix or class fd object [required].
palette_divergent	A divergent palette for the distance values [default diverging_palette].
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 heatmap.2 (package gplots)

Author(s)

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

Examples

```
## Not run:
   gl <- testset.gl[1:10,]
   D <- dist(as.matrix(gl),upper=TRUE,diag=TRUE)
   gl.plot.heatmap(D)
   D2 <- gl.dist.pop(possums.gl)
   gl.plot.heatmap(D2)
   D3 <- gl.fixed.diff(testset.gl)
   gl.plot.heatmap(D3)

## End(Not run)
   if ((requireNamespace("gplots", quietly = TRUE))) {
   D2 <- gl.dist.pop(possums.gl)
   gl.plot.heatmap(D2)
   }
```

---

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.

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.loci(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.plot.structure` *Plots STRUCTURE analysis results (Q-matrix)*

Description

This function takes a structure run object (output from `gl.run.structure`) and plots the typical structure bar plot that visualize the q matrix of a structure run.

Usage

```
gl.plot.structure(
  sr,
  K = NULL,
  met_clumpp = "greedyLargeK",
  iter_clumpp = 100,
  clumpak = TRUE,
  plot_theme = NULL,
  colors_clusters = NULL,
  ind_name = TRUE,
  border_ind = 0.15,
  plot.out = TRUE,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

`sr`	Structure run object from `gl.run.structure` [required].
`K`	The number for K of the q matrix that should be plotted. Needs to be within you simulated range of K's in your sr structure run object. If NULL, all the K's are plotted [default NULL].
`met_clumpp`	The algorithm to use to infer the correct permutations. One of 'greedy' or 'greedyLargeK' or 'stephens' [default "greedyLargeK"].
`iter_clumpp`	The number of iterations to use if running either 'greedy' 'greedyLargeK' [default 100].
`clumpak`	Whether use the Clumpak method (see details) [default TRUE].
`plot_theme`	Theme for the plot. See Details for options [default NULL].
`colors_clusters`	A color palette for clusters (K) or a list with as many colors as there are clusters (K) [default NULL].
`ind_name`	Whether to plot individual names [default TRUE].
`border_ind`	The width of the border line between individuals [default 0.25].
`plot.out`	Specify if plot is to be produced [default TRUE].

save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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 function outputs a barplot which is the typical output of structure. For a Evanno plot use gl.evanno.

This function is based on the methods of CLUMPP and Clumpak as implemented in the R package starmie (https://github.com/sa-lee/starmie).

The Clumpak method identifies sets of highly similar runs among all the replicates of the same K. The method then separates the distinct groups of runs representing distinct modes in the space of possible solutions.

The CLUMPP method permutes the clusters output by independent runs of clustering programs such as structure, so that they match up as closely as possible.

This function averages the replicates within each mode identified by the Clumpak method.

Plots and table are saved to the temporal directory (tempdir) and can be accessed with the function gl.print.reports and listed with the function gl.list.reports. Note that they can be accessed only in the current R session because tempdir is cleared each time that the R session is closed.

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

List of Q-matrices

Author(s)

Bernd Gruber & Luis Mijangos (Post to https://groups.google.com/d/forum/dartr)

References

- Pritchard, J.K., Stephens, M., Donnelly, P. (2000) Inference of population structure using multilocus genotype data. Genetics 155, 945-959.
- Kopelman, Naama M., et al. "Clumpak: a program for identifying clustering modes and packaging population structure inferences across K." Molecular ecology resources 15.5 (2015): 1179-1191.
- Mattias Jakobsson and Noah A. Rosenberg. 2007. CLUMPP: a cluster matching and permutation program for dealing with label switching and multimodality in analysis of population structure. Bioinformatics 23(14):1801-1806. Available at clumpp

See Also

gl.run.structure, gl.plot.structure

Examples

```
## Not run:
#bc <- bandicoot.gl[,1:100]
#sr <- gl.run.structure(bc, k.range = 2:5, num.k.rep = 3, exec = './structure')
#ev <- gl.evanno(sr)
#ev
#qmat <- gl.plot.structure(sr, K=3)
#head(qmat)
#gl.map.structure(qmat, K=3, bc, scalex=1, scaley=0.5)

## End(Not run)
```

---

gl.print.history *Prints history of a genlight object*

Description

Prints history of a genlight object

Usage

```
gl.print.history(x = NULL, history = NULL)
```

Arguments

x	A genlight object (with history) [optional].
history	Either a link to a history slot (gl\@other$history), or a vector indicating which part of the history of x is used [c(1,3,4) uses the first, third and forth entry from x\@other$history]. If no history is provided the complete history of x is used (recreating the identical object x) [optional].

Value

Prints a table with all history records. Currently the style cannot be changed.

Author(s)

Bernd Gruber (bugs? Post to https://groups.google.com/d/forum/dartr)

Examples

```
dartfile <- system.file('extdata','testset_SNPs_2Row.csv', package='dartR')
metadata <- system.file('extdata','testset_metadata.csv', package='dartR')
gl <- gl.read.dart(dartfile, ind.metafile = metadata, probar=FALSE)
gl2 <- gl.filter.callrate(gl, method='loc', threshold=0.9)
gl3 <- gl.filter.callrate(gl2, method='ind', threshold=0.95)
#Now 'replay' part of the history 'onto' another genlight object
```

```
#bc.fil <- gl.play.history(gl.compliance.check(bandicoot.gl),
#history=gl3@other$history[c(2,3)], verbose=1)
#gl.print.history(bc.fil)
```

`gl.print.reports` *Prints dartR reports saved in tempdir*

Description

Prints dartR reports saved in tempdir

Usage

```
gl.print.reports(print_report)
```

Arguments

`print_report`	Number of report from `gl.list.reports` that is to be printed

Value

Prints reports that were saved in tempdir.

Author(s)

Bernd Gruber & Luis Mijangos (bugs? Post to https://groups.google.com/d/forum/dartr)

See Also

`gl.list.reports`

Examples

```
## Not run:
reports <- gl.print.reports(1)

## End(Not run)
```

gl.propShared — *Calculates a similarity (distance) matrix for individuals on the proportion of shared alleles*

Description

This script calculates an individual based distance matrix. It uses an C++ implementation, so package Rcpp needs to be installed and it is therefore really fast (once it has compiled the function after the first run).

Usage

```
gl.propShared(x)
```

Arguments

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

Author(s)

Bernd Gruber (Post to https://groups.google.com/d/forum/dartr)

Examples

```
#takes some time at the first run of the function...
## Not run:
res <- gl.propShared(bandicoot.gl)
res[1:5,1:7] #show only a small part of the matrix

## End(Not run)
```

gl.random.snp — *Randomly changes the allocation of 0's and 2's in a genlight object*

Description

This function samples randomly half of the SNPs and re-codes, in the sampled SNP's, 0's by 2's.

Usage

```
gl.random.snp(x, plot.out = TRUE, save2tmp = FALSE, verbose = NULL)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
plot.out	Specify if a plot is to be produced [default TRUE].
save2tmp	If TRUE, saves any ggplots to the session temporary directory (tempdir) [default FALSE].
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

DArT calls the most common allele as the reference allele. In a genlight object, homozygous for the reference allele are coded with a '0' and homozygous for the alternative allele are coded with a '2'. This causes some distortions in visuals from time to time.

If plot.out = TRUE, two smear plots (pre-randomisation and post-randomisation) are presented using a random subset of individuals (10) and loci (100) to provide an overview of the changes.

Resultant ggplots are saved to the session's temporary directory.

Value

Returns a genlight object with half of the loci re-coded.

Author(s)

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

Examples

```
require("dartR.data")
res <- gl.random.snp(platypus.gl[1:5,1:5],verbose = 5)
```

---

gl.read.csv *Reads SNP data from a csv file into a genlight object*

Description

This script takes SNP genotypes from a csv file, combines them with individual and locus metrics and creates a genlight object.

Usage

```
gl.read.csv(
  filename,
  transpose = FALSE,
  ind.metafile = NULL,
  loc.metafile = NULL,
  verbose = NULL
)
```

Arguments

filename	Name of the csv file containing the SNP genotypes [required].
transpose	If TRUE, rows are loci and columns are individuals [default FALSE].
ind.metafile	Name of the csv file containing the metrics for individuals [optional].
loc.metafile	Name of the csv file containing the metrics for loci [optional].
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 SNP data need to be in one of two forms. SNPs can be coded 0 for homozygous reference, 2 for homozygous alternate, 1 for heterozygous, and NA for missing values; or the SNP data can be coded A/A, A/C, C/T, G/A etc, and -/- for missing data. In this format, the reference allele is the most frequent allele, as used by DArT. Other formats will throw an error.

The SNP data need to be individuals as rows, labeled, and loci as columns, also labeled. If the orientation is individuals as columns and loci by rows, then set transpose=TRUE.

The individual metrics need to be in a csv file, with headings, with a mandatory id column corresponding exactly to the individual identity labels provided with the SNP data and in the same order.

The locus metadata needs to be in a csv file with headings, with a mandatory column headed AlleleID corresponding exactly to the locus identity labels provided with the SNP data and in the same order.

Note that the locus metadata will be complemented by calculable statistics corresponding to those that would be provided by Diversity Arrays Technology (e.g. CallRate).

Value

A genlight object with the SNP data and associated metadata included.

Author(s)

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

Examples

```
csv_file <- system.file('extdata','platy_test.csv', package='dartR')
ind_metadata <- system.file('extdata','platy_ind.csv', package='dartR')
gl  <- gl.read.csv(filename = csv_file, ind.metafile = ind_metadata)
```

gl.read.dart *Imports DArT data into dartR and converts it into a genlight object*

Description

This function is a wrapper function that allows you to convert your DArT file into a genlight object in one step. In previous versions you had to use read.dart and then dart2genlight. In case you have individual metadata for each individual/sample you can specify as before in the dart2genlight command the file that combines the data.

Usage

```
gl.read.dart(
  filename,
  ind.metafile = NULL,
  recalc = TRUE,
  mono.rm = FALSE,
  nas = "-",
  topskip = NULL,
  lastmetric = "RepAvg",
  covfilename = NULL,
  service_row = 1,
  plate_row = 3,
  probar = FALSE,
  verbose = NULL
)
```

Arguments

filename	File containing the SNP data (csv file) [required].
ind.metafile	File that contains additional information on individuals [required].
recalc	Force the recalculation of locus metrics, in case individuals have been manually deleted from the input csv file [default TRUE].
mono.rm	Force the removal of monomorphic loci (including all NAs), in case individuals have been manually deleted from the input csv file [default FALSE].
nas	A character specifying NAs [default '-'].
topskip	A number specifying the number of rows to be skipped. If not provided the number of rows to be skipped are 'guessed' by the number of rows with '*' at the beginning [default NULL].
lastmetric	Specifies the last non-genetic column (Default is 'RepAvg'). Be sure to check if that is true, otherwise the number of individuals will not match. You can also specify the last column by a number [default 'RepAvg'].
covfilename	Use ind.metafile parameter [depreciated, NULL].
service_row	The row number in which the information of the DArT service is contained [default 1].

plate_row	The row number in which the information of the plate location is contained [default 3].
probar	Show progress bar [default FALSE].
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 set by gl.set.verbose()].

Details

The dartR genlight object can then be fed into a number of initial screening, export and export functions provided by the package. For some of the functions it is necessary to have the metadata that was provided from DArT. Please check the vignette for more information. Additional information can also be found in the help documents for utils.read.dart.

Value

A genlight object that contains individual metrics [if data were provided] and locus metrics [from a DArT report].

Author(s)

Custodian: Bernd Gruber (Post to https://groups.google.com/d/forum/dartr)

See Also

utils.read.dart

Examples

```
dartfile <- system.file('extdata','testset_SNPs_2Row.csv', package='dartR')
metadata <- system.file('extdata','testset_metadata.csv', package='dartR')
gl <- gl.read.dart(dartfile, ind.metafile = metadata, probar=TRUE)
```

---

gl.read.fasta *Reads FASTA files and converts them to genlight object*

Description

The following IUPAC Ambiguity Codes are taken as heterozygotes:

- M is heterozygote for AC and CA
- R is heterozygote for AG and GA
- W is heterozygote for AT and TA
- S is heterozygote for CG and GC
- Y is heterozygote for CT and TC
- K is heterozygote for GT and TG

The following IUPAC Ambiguity Codes are taken as missing data:

- V
- H
- D
- B
- N

The function can deal with missing data in individuals, e.g. when FASTA files have different number of individuals due to missing data.

The allele with the highest frequency is taken as the reference allele.

SNPs with more than two alleles are skipped.

Usage

```
gl.read.fasta(fasta_files, parallel = FALSE, n_cores = NULL, verbose = NULL)
```

Arguments

`fasta_files`	Fasta files to read [required].
`parallel`	A logical indicating whether multiple cores -if available- should be used for the computations (TRUE), or not (FALSE); requires the package parallel to be installed [default FALSE].
`n_cores`	If parallel is TRUE, the number of cores to be used in the computations; if NULL, then the maximum number of cores available on the computer is used [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, unless specified using gl.set.verbosity].

Details

Ambiguity characters are often used to code heterozygotes. However, using heterozygotes as ambiguity characters may bias many estimates. See more information in the link below: https://evodify.com/heterozygotes-ambiguity-characters/

Value

A genlight object.

Author(s)

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

Examples

```
# Folder where the fasta files are located.
folder_samples <- system.file('extdata', package ='dartR')
# listing the FASTA files, including their path. Files have an extension
# that contains "fas".
file_names <- list.files(path = folder_samples, pattern = "*.fas",
                         full.names = TRUE)
# reading fasta files
 obj <- gl.read.fasta(file_names)
```

`gl.read.silicodart` *Imports presence/absence data from SilicoDArT to genlight {adegenet} format (ploidy=1)*

Description

DaRT provide the data as a matrix of entities (individual animals) across the top and attributes (P/A of sequenced fragment) down the side in a format that is unique to DArT. This program reads the data in to adegenet format for consistency with other programming activity. The script may require modification as DArT modify their data formats from time to time.

Usage

```
gl.read.silicodart(
  filename,
  ind.metafile = NULL,
  nas = "-",
  topskip = NULL,
  lastmetric = "Reproducibility",
  probar = TRUE,
  verbose = NULL
)
```

Arguments

`filename`	Name of csv file containing the SilicoDArT data [required].
`ind.metafile`	Name of csv file containing metadata assigned to each entity (individual) [default NULL].
`nas`	Missing data character [default '-'].
`topskip`	Number of rows to skip before the header row (containing the specimen identities) [optional].
`lastmetric`	Specifies the last non genetic column (Default is 'Reproducibility'). Be sure to check if that is true, otherwise the number of individuals will not match. You can also specify the last column by a number [default "Reproducibility"].
`probar`	Show progress bar [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 set by gl.set.verbose()].

Details

gl.read.silicodart() opens the data file (csv comma delimited) and skips the first n=topskip lines. The script assumes that the next line contains the entity labels (specimen ids) followed immediately by the SNP data for the first locus.

It reads the presence/absence data into a matrix of 1s and 0s, and inputs the locus metadata and specimen metadata. The locus metadata comprises a series of columns of values for each locus including the essential columns of CloneID and the desirable variables Reproducibility and PIC. Refer to documentation provide by DArT for an explanation of these columns.

The specimen metadata provides the opportunity to reassign specimens to populations, and to add other data relevant to the specimen. The key variables are id (specimen identity which must be the same and in the same order as the SilicoDArT file, each unique), pop (population assignment), lat (latitude, optional) and lon (longitude, optional). id, pop, lat, lon are the column headers in the csv file. Other optional columns can be added.

The data matrix, locus names (forced to be unique), locus metadata, specimen names, specimen metadata are combined into a genind object. Refer to the documentation for {adegenet} for further details.

Value

An object of class `genlight` with ploidy set to 1, containing the presence/absence data, and locus and individual metadata.

Author(s)

Custodian: Bernd Gruber – Post to https://groups.google.com/d/forum/dartr

See Also

`gl.read.dart`

Examples

```
silicodartfile <- system.file('extdata','testset_SilicoDArT.csv', package='dartR')
metadata <- system.file('extdata',ind.metafile ='testset_metadata_silicodart.csv', package='dartR')
testset.gs <- gl.read.silicodart(filename = silicodartfile, ind.metafile = metadata)
```

---

gl.read.vcf *Converts a vcf file into a genlight object*

Description

This function needs package vcfR, please install it. The converted genlight object does not have individual metrics. You need to add them 'manually' to the other$ind.metrics slot.

Usage

```
gl.read.vcf(vcffile, verbose = NULL)
```

Arguments

vcffile	A vcf file (works only for diploid data) [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].

Value

A genlight object.

Author(s)

Bernd Gruber (Post to https://groups.google.com/d/forum/dartr)

Examples

```
## Not run:
obj <- gl.read.vcf(system.file('extdata/test.vcf', package='dartR'))

## End(Not run)
```

---

gl.reassign.pop *Assigns an individual metric as pop in a genlight {adegenet} object*

Description

Individuals are assigned to populations based on the individual/sample/specimen metrics file (csv) used with gl.read.dart().

One might want to define the population structure in accordance with another classification, such as using an individual metric (e.g. sex, male or female). This script discards the current population assignments and replaces them with new population assignments defined by a specified individual metric.

The script returns a genlight object with the new population assignments. Note that the original population assignments are lost.

Usage

```
gl.reassign.pop(x, as.pop, verbose = NULL)
```

Arguments

x	Name of the genlight object containing SNP genotypes [required].
as.pop	Specify the name of the individual metric to set as the pop variable [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].

Value

A genlight object with the reassigned populations.

Author(s)

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

Examples

```
# SNP data
   popNames(testset.gl)
   gl <- gl.reassign.pop(testset.gl, as.pop='sex',verbose=3)
   popNames(gl)
# Tag P/A data
   popNames(testset.gs)
   gs <- gl.reassign.pop(testset.gs, as.pop='sex',verbose=3)
   popNames(gs)
```

---

gl.recalc.metrics *Recalculates locus metrics when individuals or populations are deleted from a genlight {adegenet} object*

Description

When individuals,or populations, are deleted from a genlight object, the locus metrics no longer apply. For example, the Call Rate may be different considering the subset of individuals, compared with the full set. This script recalculates those affected locus metrics, namely, avgPIC, CallRate, freqHets, freqHomRef, freqHomSnp, OneRatioRef, OneRatioSnp, PICRef and PICSnp. Metrics that remain unaltered are RepAvg and TrimmedSeq as they are unaffected by the removal of individuals.

Usage

```
gl.recalc.metrics(x, mono.rm = FALSE, verbose = NULL)
```

Arguments

x	Name of the genlight object containing SNP genotypes [required].
mono.rm	If TRUE, removes monomorphic loci [default FALSE].
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 script optionally removes resultant monomorphic loci or loci with all values missing and deletes them (using gl.filter.monomorphs.r).

The script returns a genlight object with the recalculated locus metadata.

Value

A genlight object with the recalculated locus metadata.

Author(s)

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

See Also

gl.filter.monomorphs

Examples

```
gl <- gl.recalc.metrics(testset.gl, verbose=2)
```

---

gl.recode.ind *Recodes individual (=specimen = sample) labels in a genlight object*

Description

This script recodes individual labels and/or deletes individuals from a DaRT genlight SNP file based on a lookup table provided as a csv file.

Usage

```
gl.recode.ind(x, ind.recode, recalc = FALSE, mono.rm = FALSE, verbose = NULL)
```

Arguments

x	Name of the genlight object containing SNP genotypes [required].
ind.recode	Name of the csv file containing the individual relabelling [required].
recalc	If TRUE, recalculate the locus metadata statistics if any individuals are deleted in the filtering [default FALSE].
mono.rm	If TRUE, remove monomorphic loci [default FALSE].
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

Renaming individuals may be required when there have been errors in labelling arising in the process from sample to DArT files. There may be occasions where renaming individuals is required for preparation of figures. When caution needs to be exercised because of the potential for breaking the 'chain of evidence' associated with the samples, recoding individuals using a recode table (csv) can provide a clear record of the changes.

The script, having deleted individuals, optionally identifies resultant monomorphic loci or loci with all values missing and deletes them (using gl.filter.monomorphs.r). The script also optionally recalculates statistics made incorrect by the deletion of individuals from the dataset.

The script returns a genlight object with the new individual labels, the monomorphic loci optionally removed and the optionally recalculated locus metadata.

Value

A genlight or genind object with the recoded and reduced data.

Author(s)

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

See Also

gl.filter.monomorphs for filtering monomorphs, gl.recalc.metrics for recalculating locus metrics, gl.recode.pop for recoding populations

Examples

```
file <- system.file('extdata','testset_ind_recode.csv', package='dartR')
gl <- gl.recode.ind(testset.gl, ind.recode=file, verbose=3)
```

---

gl.recode.pop *Recodes population assignments in a genlight object*

Description

This script recodes population assignments and/or deletes populations from a DaRT genlight SNP file based on information provided in a csv population recode file.

Usage

```
gl.recode.pop(x, pop.recode, recalc = FALSE, mono.rm = FALSE, verbose = NULL)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
pop.recode	Name of the csv file containing the population reassignments [required].
recalc	Recalculate the locus metadata statistics if any individuals are deleted in the filtering [default FALSE].
mono.rm	Remove monomorphic loci [default FALSE].
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

Individuals are assigned to populations based on the specimen metadata data file (csv) used with gl.read.dart(). Recoding can be used to amalgamate populations or to selectively delete or retain populations.

The population recode file contains a list of populations in the genlight object as the first column of the csv file, and the new population assignments in the second column of the csv file. The keyword Delete used as a new population assignment will result in the associated specimen being dropped from the dataset.

The script, having deleted populations, optionally identifies resultant monomorphic loci or loci with all values missing and deletes them (using gl.filter.monomorphs.r). The script also optionally recalculates the locus metadata as appropriate.

Value

A genlight object with the recoded and reduced data.

Author(s)

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

See Also

gl.filter.monomorphs

gl.recode.pop

Examples

```
mfile <- system.file('extdata', 'testset_pop_recode.csv', package='dartR')
nPop(testset.gl)
gl <- gl.recode.pop(testset.gl, pop.recode=mfile, verbose=3)
```

---

gl.rename.pop *Renames a population in a genlight object*

Description

Individuals are assigned to populations based on the specimen metadata data file (csv) used with gl.read.dart().

This script renames a nominated population.

The script returns a genlight object with the new population name.

Usage

```
gl.rename.pop(x, old = NULL, new = NULL, verbose = NULL)
```

Arguments

x	Name of the genlight object containing SNP genotypes [required].
old	Name of population to be changed [required].
new	New name for the 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 or as specified using gl.set.verbosity].

Value

A genlight object with the new population name.

Author(s)

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

Examples

```
gl <- gl.rename.pop(testset.gl, old='EmsubRopeMata', new='Outgroup')
```

---

gl.report.bases *Reports summary of base pair frequencies*

Description

This script calculates the frequencies of the four DNA nucleotide bases: adenine (A), cytosine (C), 'guanine (G) and thymine (T), and the frequency of transitions (Ts) and transversions (Tv) in a DArT genlight object.

Usage

```
gl.report.bases(
  x,
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = two_colors,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP or presence/absence (Silico-DArT) data [required].
plot.out	If TRUE, histograms of base composition are produced [default TRUE].
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 two_colors].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE]
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 script checks first if trimmed sequences are included in the locus metadata (@other$loc.metrics$TrimmedSequence), and if so, tallies up the numbers of A, T, G and C bases. Only the reference state at the SNP locus is counted. Counts of transitions (Ts) and transversions (Tv) assume that there is no directionality, that is C->T is the same as T->C, because the reference state is arbitrary.

For presence/absence data (SilicoDArT), it is not possible to count transversions or transitions or transversions/transitions ratio because the SNP data is not available, only a single sequence tag.

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 unchanged genlight object

Author(s)

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

See Also

Other report functions: gl.report.callrate(), gl.report.diversity(), gl.report.hamming(), gl.report.heterozygosity(), gl.report.hwe(), gl.report.ld.map(), gl.report.locmetric(), gl.report.maf(), gl.report.monomorphs(), gl.report.overshoot(), gl.report.parent.offspring(), gl.report.pa(), gl.report.rdepth(), gl.report.reproducibility(), gl.report.secondaries(), gl.report.sexlinked(), gl.report.taglength()

Examples

```
# SNP data
  out <- gl.report.bases(testset.gl)
  #' # Tag P/A data
  out <- gl.report.bases(testset.gs)
```

`gl.report.callrate` *Reports summary of Call Rate for loci or individuals*

Description

SNP datasets generated by DArT have missing values primarily arising from failure to call a SNP because of a mutation at one or both of the restriction enzyme recognition sites. P/A datasets (SilicoDArT) have missing values because it was not possible to call whether a sequence tag was amplified or not. This function tabulates the number of missing values as quantiles.

Usage

```
gl.report.callrate(
  x,
  method = "loc",
  by_pop = FALSE,
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = two_colors,
  bins = 50,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object containing the SNP or presence/absence (SilicoDArT) data [required].
`method`	Specify the type of report by locus (method='loc') or individual (method='ind') [default 'loc'].
`by_pop`	Whether report by population [default FALSE].
`plot.out`	Specify if plot is to be produced [default TRUE].
`plot_theme`	User specified theme [default theme_dartR()].
`plot_colors`	Vector with two color names for the borders and fill [default two_colors].
`bins`	Number of bins to display in histograms [default 25].
`save2tmp`	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
`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 expects a genlight object, containing either SNP data or SilicoDArT (=presence/absence data).

Callrate is summarized by locus or by individual to allow sensible decisions on thresholds for filtering taking into consideration consequential loss of data. The summary is in the form of a tabulation and plots.

Plot themes can be obtained from:

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

Resultant ggplots and the tabulation are saved to the session's temporary directory.

Value

Returns unaltered genlight object

Author(s)

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

See Also

gl.filter.callrate

Other report functions: gl.report.bases(), gl.report.diversity(), gl.report.hamming(), gl.report.heterozygosity(), gl.report.hwe(), gl.report.ld.map(), gl.report.locmetric(), gl.report.maf(), gl.report.monomorphs(), gl.report.overshoot(), gl.report.parent.offspring(), gl.report.pa(), gl.report.rdepth(), gl.report.reproducibility(), gl.report.secondaries(), gl.report.sexlinked(), gl.report.taglength()

Examples

```
# SNP data
  test.gl <- testset.gl[1:20,]
  gl.report.callrate(test.gl)
  gl.report.callrate(test.gl,method='ind')
# Tag P/A data
  test.gs <- testset.gs[1:20,]
  gl.report.callrate(test.gs)
  gl.report.callrate(test.gs,method='ind')

  test.gl <- testset.gl[1:20,]
  gl.report.callrate(test.gl)
```

gl.report.diversity *Calculates diversity indexes for SNPs*

Description

This script takes a genlight object and calculates alpha and beta diversity for q = 0:2. Formulas are taken from Sherwin et al. 2017. The paper describes nicely the relationship between the different q levels and how they relate to population genetic processes such as dispersal and selection.

Usage

```
gl.report.diversity(
  x,
  plot.out = TRUE,
  pbar = TRUE,
  table = "DH",
  plot_theme = theme_dartR(),
  plot_colors = discrete_palette,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP or presence/absence (Silico-DArT) data [required].
plot.out	Specify if plot is to be produced [default TRUE].
pbar	Report on progress. Silent if set to FALSE [default TRUE].
table	Prints a tabular output to the console either 'D'=D values, or 'H'=H values or 'DH','HD'=both or 'N'=no table. [default 'DH'].
plot_theme	Theme for the plot. See Details for options [default theme_dartR()].
plot_colors	A color palette or a list with as many colors as there are populations in the dataset [default discrete_palette].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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

For all indexes, the entropies (H) and corresponding effective numbers, i.e. Hill numbers (D), which reflect the number of needed entities to get the observed values, are calculated. In a nutshell, the alpha indexes between the different q-values should be similar if there is no deviation from expected allele frequencies and occurrences (e.g. all loci in HWE & equilibrium). If there is a deviation of an index, this links to a process causing it, such as dispersal, selection or strong drift. For a detailed

explanation of all the indexes, we recommend resorting to the literature provided below. Confidence intervals are +/- 1 standard deviation.

Function's output

If the function's parameter "table" = "DH" (the default value) is used, the output of the function is 20 tables.

The first two show the number of loci used. The name of each of the rest of the tables starts with three terms separated by underscores.

The first term refers to the q value (0 to 2).

The second term refers to whether it is the diversity measure (H) or its transformation to Hill numbers (D).

The third term refers to whether the diversity is calculated within populations (alpha) or between populations (beta).

In the case of alpha diversity tables, standard deviations have their own table, which finishes with a fourth term: "sd".

In the case of beta diversity tables, standard deviations are in the upper triangle of the matrix and diversity values are in the lower triangle of the matrix.

Plots are saved to the temporal directory (tempdir) and can be accessed with the function `gl.print.reports` and listed with the function `gl.list.reports`. Note that they can be accessed only in the current R session because tempdir is cleared each time that the R session is closed.

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 list of entropy indexes for each level of q and equivalent numbers for alpha and beta diversity.

Author(s)

Bernd Gruber (Post to https://groups.google.com/d/forum/dartr), Contributors: William B. Sherwin, Alexander Sentinella

References

Sherwin, W.B., Chao, A., Johst, L., Smouse, P.E. (2017). Information Theory Broadens the Spectrum of Molecular Ecology and Evolution. TREE 32(12) 948-963. doi:10.1016/j.tree.2017.09.12

See Also

Other report functions: `gl.report.bases()`, `gl.report.callrate()`, `gl.report.hamming()`, `gl.report.heterozygosity()`, `gl.report.hwe()`, `gl.report.ld.map()`, `gl.report.locmetric()`, `gl.report.maf()`, `gl.report.monomorphs()`, `gl.report.overshoot()`, `gl.report.parent.offspring()`, `gl.report.pa()`, `gl.report.rdepth()`, `gl.report.reproducibility()`, `gl.report.secondaries()`, `gl.report.sexlinked()`, `gl.report.taglength()`

Examples

```
div <- gl.report.diversity(bandicoot.gl[1:10,1:100], table = FALSE,
 pbar=FALSE)
div$zero_H_alpha
div$two_H_beta
names(div)
```

`gl.report.hamming` *Calculates the pairwise Hamming distance between DArT trimmed DNA sequences*

Description

Hamming distance is calculated as the number of base differences between two sequences which can be expressed as a count or a proportion. Typically, it is calculated between two sequences of equal length. In the context of DArT trimmed sequences, which differ in length but which are anchored to the left by the restriction enzyme recognition sequence, it is sensible to compare the two trimmed sequences starting from immediately after the common recognition sequence and terminating at the last base of the shorter sequence.

Usage

```
gl.report.hamming(
  x,
  rs = 5,
  threshold = 3,
  taglength = 69,
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = two_colors,
  probar = FALSE,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object containing the SNP data [required].
`rs`	Number of bases in the restriction enzyme recognition sequence [default 5].
`threshold`	Minimum acceptable base pair difference for display on the boxplot and histogram [default 3].
`taglength`	Typical length of the sequence tags [default 69].
`plot.out`	Specify if plot is to be produced [default TRUE].
`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 two_colors].
probar	If TRUE, then a progress bar is displayed on long loops [default TRUE].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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 function gl.filter.hamming will filter out one of two loci if their Hamming distance is less than a specified percentage

Hamming distance can be computed by exploiting the fact that the dot product of two binary vectors x and (1-y) counts the corresponding elements that are different between x and y. This approach can also be used for vectors that contain more than two possible values at each position (e.g. A, C, T or G).

If a pair of DNA sequences are of differing length, the longer is truncated.

The algorithm is that of Johann de Jong https://johanndejong.wordpress.com/2015/10/02/faster-hamming-distance-in-r-2/ as implemented in utils.hamming

Plots and table are saved to the session's temporary directory (tempdir)

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

Returns unaltered genlight object

Author(s)

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

See Also

gl.filter.hamming

Other report functions: gl.report.bases(), gl.report.callrate(), gl.report.diversity(), gl.report.heterozygosity(), gl.report.hwe(), gl.report.ld.map(), gl.report.locmetric(), gl.report.maf(), gl.report.monomorphs(), gl.report.overshoot(), gl.report.parent.offspring(), gl.report.pa(), gl.report.rdepth(), gl.report.reproducibility(), gl.report.secondaries(), gl.report.sexlinked(), gl.report.taglength()

Examples

```
gl.report.hamming(testset.gl[,1:100])
gl.report.hamming(testset.gs[,1:100])
```

```
#' # SNP data
test <- platypus.gl
test <- gl.subsample.loci(platypus.gl,n=50)
result <- gl.filter.hamming(test, threshold=0.25, verbose=3)
```

---

gl.report.heterozygosity

Reports observed, expected and unbiased heterozygosities and FIS (inbreeding coefficient) by population or by individual from SNP data

---

Description

Calculates the observed, expected and unbiased expected (i.e. corrected for sample size) heterozygosities and FIS (inbreeding coefficient) for each population or the observed heterozygosity for each individual in a genlight object.

Usage

```
gl.report.heterozygosity(
  x,
  method = "pop",
  n.invariant = 0,
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors_pop = discrete_palette,
  plot_colors_ind = two_colors,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP [required].
method	Calculate heterozygosity by population (method='pop') or by individual (method='ind') [default 'pop'].
n.invariant	An estimate of the number of invariant sequence tags used to adjust the heterozygosity rate [default 0].
plot.out	Specify if plot is to be produced [default TRUE].
plot_theme	Theme for the plot. See Details for options [default theme_dartR()].
plot_colors_pop	A color palette for population plots or a list with as many colors as there are populations in the dataset [default discrete_palette].
plot_colors_ind	List of two color names for the borders and fill of the plot by individual [default two_colors].

save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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

Observed heterozygosity for a population takes the proportion of heterozygous loci for each individual then averages over the individuals in that population. The calculations take into account missing values.

Expected heterozygosity for a population takes the expected proportion of heterozygotes, that is, expected under Hardy-Weinberg equilibrium, for each locus, then averages this across the loci for an average estimate for the population.

Observed heterozygosity for individuals is calculated as the proportion of loci that are heterozygous for that individual.

Finally, the loci that are invariant across all individuals in the dataset (that is, across populations), is typically unknown. This can render estimates of heterozygosity analysis specific, and so it is not valid to compare such estimates across species or even across different analyses. This is a similar problem faced by microsatellites. If you have an estimate of the number of invariant sequence tags (loci) in your data, such as provided by `gl.report.secondaries`, you can specify it with the n.invariant parameter to standardize your estimates of heterozygosity.

NOTE: It is important to realise that estimation of adjusted heterozygosity requires that secondaries not to be removed.

Heterozygosities and FIS (inbreeding coefficient) are calculated by locus within each population using the following equations:

- Observed heterozygosity (Ho) = number of homozygotes / n_Ind, where n_Ind is the number of individuals without missing data.
- Observed heterozygosity adjusted (Ho.adj) <- Ho * n_Loc / (n_Loc + n.invariant), where n_Loc is the number of loci that do not have all missing data and n.invariant is an estimate of the number of invariant loci to adjust heterozygosity.
- Expected heterozygosity (He) = 1 - (p^2 + q^2), where p is the frequency of the reference allele and q is the frequency of the alternative allele.
- Expected heterozygosity adjusted (He.adj) = He * n_Loc / (n_Loc + n.invariant)
- Unbiased expected heterozygosity (uHe) = He * (2 * n_Ind / (2 * n_Ind - 1))
- Inbreeding coefficient (FIS) = 1 - (mean(Ho) / mean(uHe))

Function's output

Output for method='pop' is an ordered barchart of observed heterozygosity, expected heterozygosity and FIS (Inbreeding coefficient) across populations together with a table of mean observed and expected heterozygosities and FIS by population and their respective standard deviations (SD).

In the output, it is also reported by population: the number of loci used to estimate heterozygosity(nLoc), the number of polymorphic loci (polyLoc), the number of monomorphic loci (monoLoc) and loci with all missing data (all_NALoc).

Output for method='ind' is a histogram and a boxplot of heterozygosity across individuals.

Plots and table are saved to the session temporary directory (tempdir)

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 dataframe containing population labels, heterozygosities, FIS, their standard deviations and sample sizes

Author(s)

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

See Also

gl.filter.heterozygosity

Other report functions: gl.report.bases(), gl.report.callrate(), gl.report.diversity(), gl.report.hamming(), gl.report.hwe(), gl.report.ld.map(), gl.report.locmetric(), gl.report.maf(), gl.report.monomorphs(), gl.report.overshoot(), gl.report.parent.offspring(), gl.report.pa(), gl.report.rdepth(), gl.report.reproducibility(), gl.report.secondaries(), gl.report.sexlinked(), gl.report.taglength()

Examples

```
require("dartR.data")
df <- gl.report.heterozygosity(platypus.gl)
df <- gl.report.heterozygosity(platypus.gl,method='ind')
n.inv <- gl.report.secondaries(platypus.gl)
gl.report.heterozygosity(platypus.gl, n.invariant = n.inv[7, 2])

df <- gl.report.heterozygosity(platypus.gl)
```

---

gl.report.hwe *Reports departure from Hardy-Weinberg proportions*

Description

Calculates the probabilities of agreement with H-W proportions based on observed frequencies of reference homozygotes, heterozygotes and alternate homozygotes.

Usage

```
gl.report.hwe(
  x,
  subset = "each",
  method_sig = "Exact",
  multi_comp = FALSE,
  multi_comp_method = "BY",
  alpha_val = 0.05,
  pvalue_type = "midp",
  cc_val = 0.5,
  sig_only = TRUE,
  min_sample_size = 5,
  plot.out = TRUE,
  plot_colors = two_colors_contrast,
  max_plots = 4,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
subset	Way to group individuals to perform H-W tests. Either a vector with population names, 'each', 'all' (see details) [default 'each'].
method_sig	Method for determining statistical significance: 'ChiSquare' or 'Exact' [default 'Exact'].
multi_comp	Whether to adjust p-values for multiple comparisons [default FALSE].
multi_comp_method	Method to adjust p-values for multiple comparisons: 'holm', 'hochberg', 'hommel', 'bonferroni', 'BH', 'BY', 'fdr' (see details) [default 'fdr'].
alpha_val	Level of significance for testing [default 0.05].
pvalue_type	Type of p-value to be used in the Exact method. Either 'dost','selome','midp' (see details) [default 'midp'].
cc_val	The continuity correction applied to the ChiSquare test [default 0.5].
sig_only	Whether the returned table should include loci with a significant departure from Hardy-Weinberg proportions [default TRUE].
min_sample_size	Minimum number of individuals per population in which perform H-W tests [default 5].
plot.out	If TRUE, will produce Ternary Plot(s) [default TRUE].
plot_colors	Vector with two color names for the significant and not-significant loci [default two_colors_contrast].
max_plots	Maximum number of plots to print per page [default 4].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].

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

There are several factors that can cause deviations from Hardy-Weinberg proportions including: mutation, finite population size, selection, population structure, age structure, assortative mating, sex linkage, nonrandom sampling and genotyping errors. Therefore, testing for Hardy-Weinberg proportions should be a process that involves a careful evaluation of the results, a good place to start is Waples (2015).

Note that tests for H-W proportions are only valid if there is no population substructure (assuming random mating) and have sufficient power only when there is sufficient sample size (n individuals > 15).

Populations can be defined in three ways:

- Merging all populations in the dataset using subset = 'all'.
- Within each population separately using: subset = 'each'.
- Within selected populations using for example: subset = c('pop1','pop2').

Two different statistical methods to test for deviations from Hardy Weinberg proportions:

- The classical chi-square test (method_sig='ChiSquare') based on the function HWChisq of the R package HardyWeinberg. By default a continuity correction is applied (cc_val=0.5). The continuity correction can be turned off (by specifying cc_val=0), for example in cases of extreme allele frequencies in which the continuity correction can lead to excessive type 1 error rates.
- The exact test (method_sig='Exact') based on the exact calculations contained in the function HWExactStats of the R package HardyWeinberg, and described in Wigginton et al. (2005). The exact test is recommended in most cases (Wigginton et al., 2005). Three different methods to estimate p-values (pvalue_type) in the Exact test can be used:
  - 'dost' p-value is computed as twice the tail area of a one-sided test.
  - 'selome' p-value is computed as the sum of the probabilities of all samples less or equally likely as the current sample.
  - 'midp', p-value is computed as half the probability of the current sample + the probabilities of all samples that are more extreme.

  The standard exact p-value is overly conservative, in particular for small minor allele frequencies. The mid p-value ameliorates this problem by bringing the rejection rate closer to the nominal level, at the price of occasionally exceeding the nominal level (Graffelman & Moreno, 2013).

Correction for multiple tests can be applied using the following methods based on the function p.adjust:

- 'holm' is also known as the sequential Bonferroni technique (Rice, 1989). This method has a greater statistical power than the standard Bonferroni test, however this method becomes very stringent when many tests are performed and many real deviations from the null hypothesis can go undetected (Waples, 2015).
- 'hochberg' based on Hochberg, 1988.

- 'hommel' based on Hommel, 1988. This method is more powerful than Hochberg's, but the difference is usually small.
- 'bonferroni' in which p-values are multiplied by the number of tests. This method is very stringent and therefore has reduced power to detect multiple departures from the null hypothesis.
- 'BH' based on Benjamini & Hochberg, 1995.
- 'BY' based on Benjamini & Yekutieli, 2001.

The first four methods are designed to give strong control of the family-wise error rate. The last two methods control the false discovery rate (FDR), the expected proportion of false discoveries among the rejected hypotheses. The false discovery rate is a less stringent condition than the family-wise error rate, so these methods are more powerful than the others, especially when number of tests is large. The number of tests on which the adjustment for multiple comparisons is the number of populations times the number of loci.

Ternary plots

Ternary plots can be used to visualise patterns of H-W proportions (plot.out = TRUE). P-values and the statistical (non)significance of a large number of bi-allelic markers can be inferred from their position in a ternary plot. See Graffelman & Morales-Camarena (2008) for further details. Ternary plots are based on the function HWTernaryPlot from the package HardyWeinberg. Each vertex of the Ternary plot represents one of the three possible genotypes for SNP data: homozygous for the reference allele (AA), heterozygous (AB) and homozygous for the alternative allele (BB). Loci deviating significantly from Hardy-Weinberg proportions after correction for multiple tests are shown in pink. The blue parabola represents Hardy-Weinberg equilibrium, and the area between green lines represents the acceptance region.

For these plots to work it is necessary to install the package ggtern.

Value

A dataframe containing loci, counts of reference SNP homozygotes, heterozygotes and alternate SNP homozygotes; probability of departure from H-W proportions, per locus significance with and without correction for multiple comparisons and the number of population where the same locus is significantly out of HWE.

Author(s)

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

References

- Benjamini, Y., and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. Annals of Statistics, 29, 1165–1188.
- Graffelman, J. (2015). Exploring Diallelic Genetic Markers: The Hardy Weinberg Package. Journal of Statistical Software 64:1-23.
- Graffelman, J. & Morales-Camarena, J. (2008). Graphical tests for Hardy-Weinberg equilibrium based on the ternary plot. Human Heredity 65:77-84.
- Graffelman, J., & Moreno, V. (2013). The mid p-value in exact tests for Hardy-Weinberg equilibrium. Statistical applications in genetics and molecular biology, 12(4), 433-448.

- Hochberg, Y. (1988). A sharper Bonferroni procedure for multiple tests of significance. Biometrika, 75, 800–803.
- Hommel, G. (1988). A stagewise rejective multiple test procedure based on a modified Bonferroni test. Biometrika, 75, 383–386.
- Rice, W. R. (1989). Analyzing tables of statistical tests. Evolution, 43(1), 223-225.
- Waples, R. S. (2015). Testing for Hardy–Weinberg proportions: have we lost the plot?. Journal of heredity, 106(1), 1-19.
- Wigginton, J.E., Cutler, D.J., & Abecasis, G.R. (2005). A Note on Exact Tests of Hardy-Weinberg Equilibrium. American Journal of Human Genetics 76:887-893.

See Also

gl.filter.hwe

Other report functions: gl.report.bases(), gl.report.callrate(), gl.report.diversity(), gl.report.hamming(), gl.report.heterozygosity(), gl.report.ld.map(), gl.report.locmetric(), gl.report.maf(), gl.report.monomorphs(), gl.report.overshoot(), gl.report.parent.offspring(), gl.report.pa(), gl.report.rdepth(), gl.report.reproducibility(), gl.report.secondaries(), gl.report.sexlinked(), gl.report.taglength()

---

gl.report.ld *Calculates pairwise population based Linkage Disequilibrium across all loci using the specified number of cores*

Description

This function is implemented in a parallel fashion to speed up the process. There is also the ability to restart the function if crashed by specifying the chunk file names or restarting the function exactly in the same way as in the first run. This is implemented because sometimes, due to connectivity loss between cores, the function may crash half way. Before running the function, it is advisable to use the function gl.filter.allna to remove loci with all missing data.

Usage

```
gl.report.ld(
  x,
  name = NULL,
  save = TRUE,
  outpath = tempdir(),
  nchunks = 2,
  ncores = 1,
  chunkname = NULL,
  probar = FALSE,
  verbose = NULL
)
```

Arguments

x	A genlight or genind object created (genlight objects are internally converted via gl2gi to genind) [required].
name	Character string for rdata file. If not given genind object name is used [default NULL].
save	Switch if results are saved in a file [default TRUE].
outpath	Folder where chunks and results are saved (if save=TRUE) [default tempdir()].
nchunks	How many subchunks will be used (the less the faster, but if the routine crashes more bits are lost) [default 2].
ncores	How many cores should be used [default 1].
chunkname	The name of the chunks for saving [default NULL].
probar	if TRUE, a progress bar is displayed for long loops [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].

Value

Returns calculation of pairwise LD across all loci between subpopulations. This functions uses if specified many cores on your computer to speed up. And if save is used can restart (if save=TRUE is used) with the same command starting where it crashed. The final output is a data frame that holds all statistics of pairwise LD between loci. (See ?LD in package genetics for details).

Author(s)

Bernd Gruber (Post to https://groups.google.com/d/forum/dartr)

---

gl.report.ld.map *Calculates pairwise linkage disequilibrium by population*

Description

This function calculates pairwise linkage disequilibrium (LD) by population using the function ld (package snpStats).

If SNPs are not mapped to a reference genome, the parameter ld_max_pairwise should be set as NULL (the default). In this case, the function will assign the same chromosome ("1") to all the SNPs in the dataset and assign a sequence from 1 to n loci as the position of each SNP. The function will then calculate LD for all possible SNP pair combinations.

If SNPs are mapped to a reference genome, the parameter ld_max_pairwise should be filled out (i.e. not NULL). In this case, the information for SNP's position should be stored in the genlight accessor "@position" and the SNP's chromosome name in the accessor "@chromosome" (see examples). The function will then calculate LD within each chromosome and for all possible SNP pair combinations within a distance of ld_max_pairwise.

Usage

```
gl.report.ld.map(
  x,
  ld_max_pairwise = NULL,
  maf = 0.05,
  ld_stat = "R.squared",
  ind.limit = 10,
  stat_keep = "AvgPIC",
  ld_threshold_pops = 0.2,
  plot.out = TRUE,
  plot_theme = NULL,
  histogram_colors = NULL,
  boxplot_colors = NULL,
  bins = 50,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
ld_max_pairwise	Maximum distance in number of base pairs at which LD should be calculated [default NULL].
maf	Minor allele frequency (by population) threshold to filter out loci. If a value > 1 is provided it will be interpreted as MAC (i.e. the minimum number of times an allele needs to be observed) [default 0.05].
ld_stat	The LD measure to be calculated: "LLR", "OR", "Q", "Covar", "D.prime", "R.squared", and "R". See ld (package snpStats) for details [default "R.squared"].
ind.limit	Minimum number of individuals that a population should contain to take it in account to report loci in LD [default 10].
stat_keep	Name of the column from the slot loc.metrics to be used to choose SNP to be kept [default "AvgPIC"].
ld_threshold_pops	LD threshold to report in the plot of "Number of populations in which the same SNP pair are in LD" [default 0.2].
plot.out	Specify if plot is to be produced [default TRUE].
plot_theme	User specified theme [default NULL].
histogram_colors	Vector with two color names for the borders and fill [default NULL].
boxplot_colors	A color palette for box plots by population or a list with as many colors as there are populations in the dataset [default NULL].
bins	Number of bins to display in histograms [default 50].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].

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 reports LD between SNP pairs by population. The function gl.filter.ld filters out the SNPs in LD using as input the results of gl.report.ld.map. The actual number of SNPs to be filtered out depends on the parameters set in the function gl.filter.ld.

Boxplots of LD by population and a histogram showing LD frequency are presented.

Value

A dataframe with information for each SNP pair in LD.

Author(s)

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

See Also

gl.filter.ld

Other report functions: gl.report.bases(), gl.report.callrate(), gl.report.diversity(), gl.report.hamming(), gl.report.heterozygosity(), gl.report.hwe(), gl.report.locmetric(), gl.report.maf(), gl.report.monomorphs(), gl.report.overshoot(), gl.report.parent.offspring(), gl.report.pa(), gl.report.rdepth(), gl.report.reproducibility(), gl.report.secondaries(), gl.report.sexlinked(), gl.report.taglength()

Examples

```
require("dartR.data")
x <- platypus.gl
x <- gl.filter.callrate(x,threshold = 1)
x <- gl.filter.monomorphs(x)
x$position <- x$other$loc.metrics$ChromPos_Platypus_Chrom_NCBIv1
x$chromosome <- as.factor(x$other$loc.metrics$Chrom_Platypus_Chrom_NCBIv1)
ld_res <- gl.report.ld.map(x,ld_max_pairwise = 10000000)
```

---

gl.report.locmetric *Reports summary of the slot $other$loc.metrics*

Description

This script uses any field with numeric values stored in $other$loc.metrics to produce summary statistics (mean, minimum, average, quantiles), histograms and boxplots to assist the decision of choosing thresholds for the filter function gl.filter.locmetric.

Usage

```
gl.report.locmetric(
  x,
  metric,
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = two_colors,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP or presence/absence (Silico-DArT) data [required].
metric	Name of the metric to be used for filtering [required].
plot.out	Specify if plot is to be produced [default TRUE].
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 two_colors].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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 function gl.filter.locmetric will filter out the loci with a locmetric value below a specified threshold.

The fields that are included in dartR, and a short description, are found below. Optionally, the user can also set his/her own field by adding a vector into $other$loc.metrics as shown in the example. You can check the names of all available loc.metrics via: names(gl$other$loc.metrics).

- SnpPosition - position (zero is position 1) in the sequence tag of the defined SNP variant base.
- CallRate - proportion of samples for which the genotype call is non-missing (that is, not '-').
- OneRatioRef - proportion of samples for which the genotype score is 0.
- OneRatioSnp - proportion of samples for which the genotype score is 2.
- FreqHomRef - proportion of samples homozygous for the Reference allele.
- FreqHomSnp - proportion of samples homozygous for the Alternate (SNP) allele.
- FreqHets - proportion of samples which score as heterozygous, that is, scored as 1.
- PICRef - polymorphism information content (PIC) for the Reference allele.
- PICSnp - polymorphism information content (PIC) for the SNP.
- AvgPIC - average of the polymorphism information content (PIC) of the reference and SNP alleles.

- AvgCountRef - sum of the tag read counts for all samples, divided by the number of samples with non-zero tag read counts, for the Reference allele row.
- AvgCountSnp - sum of the tag read counts for all samples, divided by the number of samples with non-zero tag read counts, for the Alternate (SNP) allele row.
- RepAvg - proportion of technical replicate assay pairs for which the marker score is consistent.
- rdepth - read depth.

Function's output

The minimum, maximum, mean and a tabulation of quantiles of the locmetric values against thresholds rate are provided. Output also includes a boxplot and a histogram.

Quantiles are partitions of a finite set of values into q subsets of (nearly) equal sizes. In this function q = 20. Quantiles are useful measures because they are less susceptible to long-tailed distributions and outliers.

Plots and table were saved to the temporal directory (tempdir) and can be accessed with the function gl.print.reports and listed with the function gl.list.reports. Note that they can be accessed only in the current R session because tempdir is cleared each time that the R session is closed.

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

An unaltered genlight object.

Author(s)

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

See Also

gl.filter.locmetric, gl.list.reports, gl.print.reports

Other report functions: gl.report.bases(), gl.report.callrate(), gl.report.diversity(), gl.report.hamming(), gl.report.heterozygosity(), gl.report.hwe(), gl.report.ld.map(), gl.report.maf(), gl.report.monomorphs(), gl.report.overshoot(), gl.report.parent.offspring(), gl.report.pa(), gl.report.rdepth(), gl.report.reproducibility(), gl.report.secondaries(), gl.report.sexlinked(), gl.report.taglength()

Examples

```
# adding dummy data
test <- testset.gl
test$other$loc.metrics$test <- 1:nLoc(test)
# SNP data
out <- gl.report.locmetric(test,metric='test')

# adding dummy data
test.gs <- testset.gs
```

```
test.gs$other$loc.metrics$test <- 1:nLoc(test.gs)
# Tag P/A data
out <- gl.report.locmetric(test.gs,metric='test')
```

`gl.report.maf` *Reports minor allele frequency (MAF) for each locus in a SNP dataset*

Description

This script provides summary histograms of MAF for each population in the dataset and an overall histogram to assist the decision of choosing thresholds for the filter function `gl.filter.maf`

Usage

```
gl.report.maf(
  x,
  maf.limit = 0.5,
  ind.limit = 5,
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors_pop = discrete_palette,
  plot_colors_all = two_colors,
  bins = 25,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object containing the SNP data [required].
`maf.limit`	Show histograms MAF range <= maf.limit [default 0.5].
`ind.limit`	Show histograms only for populations of size greater than ind.limit [default 5].
`plot.out`	Specify if plot is to be produced [default TRUE].
`plot_theme`	Theme for the plot. See Details for options [default theme_dartR()].
`plot_colors_pop`	A color palette for population plots [default discrete_palette].
`plot_colors_all`	List of two color names for the borders and fill of the overall plot [default two_colors].
`bins`	Number of bins to display in histograms [default 25].
`save2tmp`	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
`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 function gl.filter.maf will filter out the loci with MAF below a specified threshold.

Function's output

The minimum, maximum, mean and a tabulation of MAF quantiles against thresholds rate are provided. Output also includes a boxplot and a histogram.

This function reports the MAF for each of several quantiles. Quantiles are partitions of a finite set of values into q subsets of (nearly) equal sizes. In this function q = 20. Quantiles are useful measures because they are less susceptible to long-tailed distributions and outliers.

Plots and table are saved to the temporal directory (tempdir) and can be accessed with the function gl.print.reports and listed with the function gl.list.reports. Note that they can be accessed only in the current R session because tempdir is cleared each time that the R session is closed.

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

An unaltered genlight object

Author(s)

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

See Also

gl.filter.maf, gl.list.reports, gl.print.reports

Other report functions: gl.report.bases(), gl.report.callrate(), gl.report.diversity(), gl.report.hamming(), gl.report.heterozygosity(), gl.report.hwe(), gl.report.ld.map(), gl.report.locmetric(), gl.report.monomorphs(), gl.report.overshoot(), gl.report.parent.offspring(), gl.report.pa(), gl.report.rdepth(), gl.report.reproducibility(), gl.report.secondaries(), gl.report.sexlinked(), gl.report.taglength()

Examples

```
gl <- gl.report.maf(platypus.gl)
```

`gl.report.monomorphs` *Reports monomorphic loci*

Description

This script reports the number of monomorphic loci and those with all NAs in a genlight {adegenet} object

Usage

```
gl.report.monomorphs(x, verbose = NULL)
```

Arguments

x	Name of the input genlight object [required].
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

A DArT dataset will not have monomorphic loci, but they can arise, along with loci that are scored all NA, when populations or individuals are deleted. Retaining monomorphic loci unnecessarily increases the size of the dataset and will affect some calculations.

Note that for SNP data, NAs likely represent null alleles; in tag presence/absence data, NAs represent missing values (presence/absence could not be reliably scored)

Value

An unaltered genlight object

Author(s)

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

See Also

`gl.filter.monomorphs`

Other report functions: `gl.report.bases()`, `gl.report.callrate()`, `gl.report.diversity()`, `gl.report.hamming()`, `gl.report.heterozygosity()`, `gl.report.hwe()`, `gl.report.ld.map()`, `gl.report.locmetric()`, `gl.report.maf()`, `gl.report.overshoot()`, `gl.report.parent.offspring()`, `gl.report.pa()`, `gl.report.rdepth()`, `gl.report.reproducibility()`, `gl.report.secondaries()`, `gl.report.sexlinked()`, `gl.report.taglength()`

Examples

```
# SNP data
  gl.report.monomorphs(testset.gl)
# SilicoDArT data
  gl.report.monomorphs(testset.gs)
```

---

gl.report.overshoot *Reports loci for which the SNP has been trimmed from the sequence tag along with the adaptor*

Description

This function checks the position of the SNP within the trimmed sequence tag and identifies those for which the SNP position is outside the trimmed sequence tag. This can happen, rarely, when the sequence containing the SNP resembles the adaptor.

Usage

```
gl.report.overshoot(x, save2tmp = FALSE, verbose = NULL)
```

Arguments

x	Name of the genlight object [required].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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 SNP genotype can still be used in most analyses, but functions like gl2fasta() will present challenges if the SNP has been trimmed from the sequence tag.

Resultant ggplot(s) and the tabulation(s) are saved to the session's temporary directory.

Value

An unaltered genlight object

Author(s)

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

See Also

gl.filter.overshoot

Other report functions: gl.report.bases(), gl.report.callrate(), gl.report.diversity(), gl.report.hamming(), gl.report.heterozygosity(), gl.report.hwe(), gl.report.ld.map(), gl.report.locmetric(), gl.report.maf(), gl.report.monomorphs(), gl.report.parent.offspring(), gl.report.pa(), gl.report.rdepth(), gl.report.reproducibility(), gl.report.secondaries(), gl.report.sexlinked(), gl.report.taglength()

Examples

```
gl.report.overshoot(testset.gl)
```

---

gl.report.pa *Reports private alleles (and fixed alleles) per pair of populations*

Description

This function reports private alleles in one population compared with a second population, for all populations taken pairwise. It also reports a count of fixed allelic differences and the mean absolute allele frequency differences (AFD) between pairs of populations.

Usage

```
gl.report.pa(
  x,
  x2 = NULL,
  method = "pairwise",
  loc_names = FALSE,
  plot.out = TRUE,
  font_plot = 14,
  map.interactive = FALSE,
  palette_discrete = discrete_palette,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
x2	If two separate genlight objects are to be compared this can be provided here, but they must have the same number of SNPs [default NULL].
method	Method to calculate private alleles: 'pairwise' comparison or compare each population against the rest 'one2rest' [default 'pairwise'].
loc_names	Whether names of loci with private alleles and fixed differences should reported. If TRUE, loci names are reported using a list [default FALSE].

plot.out	Specify if Sankey plot is to be produced [default TRUE].
font_plot	Numeric font size in pixels for the node text labels [default 14].
map.interactive	Specify whether an interactive map showing private alleles between populations is to be produced [default FALSE].
palette_discrete	A discrete palette for the color of populations or a list with as many colors as there are populations in the dataset [default discrete_palette].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
verbose	Verbosity: 0, silent, fatal errors only; 1, flag function begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2 or as specified using gl.set.verbosity].

Details

Note that the number of paired alleles between two populations is not a symmetric dissimilarity measure.

If no x2 is provided, the function uses the pop(gl) hierarchy to determine pairs of populations, otherwise it runs a single comparison between x and x2.

Hint: in case you want to run comparisons between individuals (assuming individual names are unique), you can simply redefine your population names with your individual names, as below:

```
pop(gl) <- indNames(gl)
```

Definition of fixed and private alleles

The table below shows the possible cases of allele frequencies between two populations (0 = homozygote for Allele 1, x = both Alleles are present, 1 = homozygote for Allele 2).

- p: cases where there is a private allele in pop1 compared to pop2 (but not vice versa)
- f: cases where there is a fixed allele in pop1 (and pop2, as those cases are symmetric)

		pop1		
		0	**x**	**1**
	0	-	p	p,f
pop2	**x**	-	-	-
	1	p,f	p	-

The absolute allele frequency difference (AFD) in this function is a simple differentiation metric displaying intuitive properties which provides a valuable alternative to FST. For details about its properties and how it is calculated see Berner (2019).

The function also reports an estimation of the lower bound of the number of undetected private alleles using the Good-Turing frequency formula, originally developed for cryptography, which estimates in an ecological context the true frequencies of rare species in a single assemblage based on an incomplete sample of individuals. The approach is described in Chao et al. (2017). For this function, the equation 2c is used. This estimate is reported in the output table as Chao1 and Chao2.

In this function a Sankey Diagram is used to visualize patterns of private alleles between populations. This diagram allows to display flows (private alleles) between nodes (populations). Their links are represented with arcs that have a width proportional to the importance of the flow (number of private alleles).

if save2temp=TRUE, resultant plot(s) and the tabulation(s) are saved to the session's temporary directory.

Value

A data.frame. Each row shows, for each pair of populations the number of individuals in each population, the number of loci with fixed differences (same for both populations) in pop1 (compared to pop2) and vice versa. Same for private alleles and finally the absolute mean allele frequency difference between loci (AFD). If loc_names = TRUE, loci names with private alleles and fixed differences are reported in a list in addition to the dataframe.

Author(s)

Custodian: Bernd Gruber – Post to https://groups.google.com/d/forum/dartr

References

- Berner, D. (2019). Allele frequency difference AFD – an intuitive alternative to FST for quantifying genetic population differentiation. Genes, 10(4), 308.
- Chao, Anne, et al. "Deciphering the enigma of undetected species, phylogenetic, and functional diversity based on Good-Turing theory." Ecology 98.11 (2017): 2914-2929.

See Also

gl.list.reports, gl.print.reports

Other report functions: gl.report.bases(), gl.report.callrate(), gl.report.diversity(), gl.report.hamming(), gl.report.heterozygosity(), gl.report.hwe(), gl.report.ld.map(), gl.report.locmetric(), gl.report.maf(), gl.report.monomorphs(), gl.report.overshoot(), gl.report.parent.offspring(), gl.report.rdepth(), gl.report.reproducibility(), gl.report.secondaries(), gl.report.sexlinked(), gl.report.taglength()

Examples

```
out <- gl.report.pa(platypus.gl)
```

---

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.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = two_colors,
  save2tmp = FALSE,
  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.out	Creates a plot that shows the sex linked markers [default TRUE].
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 two_colors].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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.

Function's output

Plots and table are saved to the temporal directory (tempdir) and can be accessed with the function gl.print.reports and listed with the function gl.list.reports. Note that they can be accessed only in the current R session because tempdir is cleared each time that the R session is closed.

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.list.reports, gl.report.rdepth , gl.print.reports,gl.report.reproducibility, gl.filter.parent.offspri

Other report functions: gl.report.bases(), gl.report.callrate(), gl.report.diversity(), gl.report.hamming(), gl.report.heterozygosity(), gl.report.hwe(), gl.report.ld.map(), gl.report.locmetric(), gl.report.maf(), gl.report.monomorphs(), gl.report.overshoot(), gl.report.pa(), gl.report.rdepth(), gl.report.reproducibility(), gl.report.secondaries(), gl.report.sexlinked(), gl.report.taglength()

Examples

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

`gl.report.rdepth` *Reports summary of Read Depth for each locus*

Description

SNP datasets generated by DArT report AvgCountRef and AvgCountSnp as counts of sequence tags for the reference and alternate alleles respectively. These can be used to back calculate Read Depth. Fragment presence/absence datasets as provided by DArT (SilicoDArT) provide Average Read Depth and Standard Deviation of Read Depth as standard columns in their report. This function reports the read depth by locus for each of several quantiles.

Usage

```
gl.report.rdepth(
  x,
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = two_colors,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object containing the SNP or presence/absence (SilicoDArT) data [required].
`plot.out`	Specify if plot is to be produced [default TRUE].
`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 two_colors].
`save2tmp`	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
`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 function displays a table of minimum, maximum, mean and quantiles for read depth against possible thresholds that might subsequently be specified in `gl.filter.rdepth`. If plot.out=TRUE, display also includes a boxplot and a histogram to guide in the selection of a threshold for filtering on read depth.

If save2tmp=TRUE, ggplots and relevant tabulations are saved to the session's temp directory (tempdir).

For examples of themes, see

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

Value

An unaltered genlight object

Author(s)

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

See Also

gl.filter.rdepth

Other report functions: gl.report.bases(), gl.report.callrate(), gl.report.diversity(), gl.report.hamming(), gl.report.heterozygosity(), gl.report.hwe(), gl.report.ld.map(), gl.report.locmetric(), gl.report.maf(), gl.report.monomorphs(), gl.report.overshoot(), gl.report.parent.offspring(), gl.report.pa(), gl.report.reproducibility(), gl.report.secondaries(), gl.report.sexlinked(), gl.report.taglength()

Examples

```
# SNP data
df <- gl.report.rdepth(testset.gl)

df <- gl.report.rdepth(testset.gs)
```

---

gl.report.reproducibility

Reports summary of RepAvg (repeatability averaged over both alleles for each locus) or reproducibility (repeatability of the scores for fragment presence/absence)

Description

SNP datasets generated by DArT have an index, RepAvg, generated by reproducing the data independently for 30 of alleles that give a repeatable result, averaged over both alleles for each locus.

In the case of fragment presence/absence data (SilicoDArT), repeatability is the percentage of scores that are repeated in the technical replicate dataset.

Usage

```
gl.report.reproducibility(
  x,
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = two_colors,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP or presence/absence (SilicoDArT) data [required].
plot.out	If TRUE, displays a plot to guide the decision on a filter threshold [default TRUE].
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 two_colors].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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 function displays a table of minimum, maximum, mean and quantiles for repeatbility against possible thresholds that might subsequently be specified in gl.filter.reproducibility.

If plot.out=TRUE, display also includes a boxplot and a histogram to guide in the selection of a threshold for filtering on repeatability.

If save2tmp=TRUE, ggplots and relevant tabulations are saved to the session's temp directory (tempdir)

For examples of themes, see:

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

Value

An unaltered genlight object

Author(s)

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

See Also

gl.filter.reproducibility

Other report functions: gl.report.bases(), gl.report.callrate(), gl.report.diversity(), gl.report.hamming(), gl.report.heterozygosity(), gl.report.hwe(), gl.report.ld.map(), gl.report.locmetric(), gl.report.maf(), gl.report.monomorphs(), gl.report.overshoot(), gl.report.parent.offspring(), gl.report.pa(), gl.report.rdepth(), gl.report.secondaries(), gl.report.sexlinked(), gl.report.taglength()

Examples

```
# SNP data
  out <- gl.report.reproducibility(testset.gl)

# Tag P/A data
  out <- gl.report.reproducibility(testset.gs)
```

---

gl.report.secondaries *Reports loci containing secondary SNPs in sequence tags and calculates number of invariant sites*

Description

SNP datasets generated by DArT include fragments with more than one SNP (that is, with secondaries). They are recorded separately with the same CloneID (=AlleleID). These multiple SNP loci within a fragment are likely to be linked, and so you may wish to remove secondaries.

This function reports statistics associated with secondaries, and the consequences of filtering them out, and provides three plots. The first is a boxplot, the second is a barplot of the frequency of secondaries per sequence tag, and the third is the Poisson expectation for those frequencies including an estimate of the zero class (no. of sequence tags with no SNP scored).

Usage

```
gl.report.secondaries(
  x,
  nsim = 1000,
  taglength = 69,
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = two_colors,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
nsim	The number of simulations to estimate the mean of the Poisson distribution [default 1000].
taglength	Typical length of the sequence tags [default 69].
plot.out	Specify if plot is to be produced [default TRUE].
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 two_colors].

save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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 function gl.filter.secondaries will filter out the loci with secondaries retaining only one sequence tag.

Heterozygosity as estimated by the function gl.report.heterozygosity is in a sense relative, because it is calculated against a background of only those loci that are polymorphic somewhere in the dataset. To allow intercompatibility across studies and species, any measure of heterozygosity needs to accommodate loci that are invariant (autosomal heterozygosity. See Schmidt et al 2021). However, the number of invariant loci are unknown given the SNPs are detected as single point mutational variants and invariant sequences are discarded, and because of the particular additional filtering pre-analysis. Modelling the counts of SNPs per sequence tag as a Poisson distribution in this script allows estimate of the zero class, that is, the number of invariant loci. This is reported, and the veracity of the estimate can be assessed by the correspondence of the observed frequencies against those under Poisson expectation in the associated graphs. The number of invariant loci can then be optionally provided to the function gl.report.heterozygosity via the parameter n.invariants.

In case the calculations for the Poisson expectation of the number of invariant sequence tags fail to converge, try to rerun the analysis with a larger nsim values.

This function now also calculates the number of invariant sites (i.e. nucleotides) of the sequence tags (if TrimmedSequence is present in x$other$loc.metrics) or estimate these by assuming that the average length of the sequence tags is 69 nucleotides. Based on the Poisson expectation of the number of invariant sequence tags, it also estimates the number of invariant sites for these to eventually provide an estimate of the total number of invariant sites.

Note, previous version of dartR would only return an estimate of the number of invariant sequence tags (not sites).

Plots are saved to the session temporary directory (tempdir).

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 data.frame with the list of parameter values

- n.total.tags Number of sequence tags in total
- n.SNPs.secondaries Number of secondary SNP loci that would be removed on filtering
- n.invariant.tags Estimated number of invariant sequence tags
- n.tags.secondaries Number of sequence tags with secondaries
- n.inv.gen Number of invariant sites in sequenced tags
- mean.len.tag Mean length of sequence tags

- n.invariant Total Number of invariant sites (including invariant sequence tags)
- k Lambda: mean of the Poisson distribution of number of SNPs in the sequence tags

Author(s)

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

References

Schmidt, T.L., Jasper, M.-E., Weeks, A.R., Hoffmann, A.A., 2021. Unbiased population heterozygosity estimates from genome-wide sequence data. Methods in Ecology and Evolution n/a.

See Also

gl.filter.secondaries,gl.report.heterozygosity, utils.n.var.invariant

Other report functions: gl.report.bases(), gl.report.callrate(), gl.report.diversity(), gl.report.hamming(), gl.report.heterozygosity(), gl.report.hwe(), gl.report.ld.map(), gl.report.locmetric(), gl.report.maf(), gl.report.monomorphs(), gl.report.overshoot(), gl.report.parent.offspring(), gl.report.pa(), gl.report.rdepth(), gl.report.reproducibility(), gl.report.sexlinked(), gl.report.taglength()

Examples

```
require("dartR.data")
test <- gl.filter.callrate(platypus.gl,threshold = 1)
n.inv <- gl.report.secondaries(test)
gl.report.heterozygosity(test, n.invariant = n.inv[7, 2])
```

---

gl.report.sexlinked *Identifies loci that are sex linked*

Description

Alleles unique to the Y or W chromosome and monomorphic on the X chromosomes will appear in the SNP dataset as genotypes that are heterozygotic in all individuals of the heterogametic sex and homozygous in all individuals of the homogametic sex. This function identifies loci with alleles that behave in this way, as putative sex specific SNP markers.

Usage

```
gl.report.sexlinked(
  x,
  sex = NULL,
  t.het = 0.1,
  t.hom = 0.1,
  t.pres = 0.1,
  plot.out = TRUE,
```

```
  plot_theme = theme_dartR(),
  plot_colors = three_colors,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP or presence/absence (SilicoDArT) data [required].
sex	Factor that defines the sex of individuals. See explanation in details [default NULL].
t.het	Tolerance in the heterogametic sex, that is t.het=0.05 means that 5% of the heterogametic sex can be homozygous and still be regarded as consistent with a sex specific marker [default 0.1].
t.hom	Tolerance in the homogametic sex, that is t.hom=0.05 means that 5% of the homogametic sex can be heterozygous and still be regarded as consistent with a sex specific marker [default 0.1].
t.pres	Tolerance in presence, that is t.pres=0.05 means that a silicodart marker can be present in either of the sexes and still be regarded as a sex-linked marker [default 0.1].
plot.out	Creates a plot that shows the heterozygosity of males and females at each loci and shaded area in which loci can be regarded as consistent with a sex specific marker [default TRUE].
plot_theme	Theme for the plot. See Details for options [default theme_dartR()].
plot_colors	List of three color names for the not sex-linked loci, for the sex-linked loci and for the area in which sex-linked loci appear [default three_colors].
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

Sex of the individuals for which sex is known with certainty can be provided via a factor (equal to the length of the number of individuals) or to be held in the variable `x@other$ind.metrics$sex`. Coding is: M for male, F for female, U or NA for unknown/missing. The script abbreviates the entries here to the first character. So, coding of 'Female' and 'Male' works as well. Character are also converted to upper cases.

' Function's output

This function creates a plot that shows the heterozygosity of males and females at each loci or SNP data or percentage of present/absent in the case of SilicoDArT data.

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

Two lists of sex-linked loci, one for XX/XY and one for ZZ/ZW systems and a plot.

Author(s)

Arthur Georges, Bernd Gruber & Floriaan Devloo-Delva (Post to https://groups.google.com/d/forum/dartr)

See Also

Other report functions: gl.report.bases(), gl.report.callrate(), gl.report.diversity(), gl.report.hamming(), gl.report.heterozygosity(), gl.report.hwe(), gl.report.ld.map(), gl.report.locmetric(), gl.report.maf(), gl.report.monomorphs(), gl.report.overshoot(), gl.report.parent.offspring(), gl.report.pa(), gl.report.rdepth(), gl.report.reproducibility(), gl.report.secondaries(), gl.report.taglength()

Examples

```
out <- gl.report.sexlinked(testset.gl)
out <- gl.report.sexlinked(testset.gs)

test <- gl.filter.callrate(platypus.gl)
test <- gl.filter.monomorphs(test)
out <- gl.report.sexlinked(test)
```

---

gl.report.taglength *Reports summary of sequence tag length across loci*

Description

SNP datasets generated by DArT typically have sequence tag lengths ranging from 20 to 69 base pairs. This function reports summary statistics of the tag lengths.

Usage

```
gl.report.taglength(
  x,
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = two_colors,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP [required].
plot.out	If TRUE, displays a plot to guide the decision on a filter threshold [default TRUE].
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 two_colors].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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 function gl.filter.taglength will filter out the loci with a tag length below a specified threshold.

Quantiles are partitions of a finite set of values into q subsets of (nearly) equal sizes. In this function q = 20. Quantiles are useful measures because they are less susceptible to long-tailed distributions and outliers.

Function's output

The minimum, maximum, mean and a tabulation of tag length quantiles against thresholds are provided. Output also includes a boxplot and a histogram to guide in the selection of a threshold for filtering on tag length.

Plots and table are saved to the temporal directory (tempdir) and can be accessed with the function gl.print.reports and listed with the function gl.list.reports. Note that they can be accessed only in the current R session because tempdir is cleared each time that the R session is closed.

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

Returns unaltered genlight object

Author(s)

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

See Also

gl.filter.taglength, gl.list.reports, gl.print.reports

Other report functions: gl.report.bases(), gl.report.callrate(), gl.report.diversity(), gl.report.hamming(), gl.report.heterozygosity(), gl.report.hwe(), gl.report.ld.map(), gl.report.locmetric(), gl.report.maf(), gl.report.monomorphs(), gl.report.overshoot(), gl.report.parent.offspring(), gl.report.pa(), gl.report.rdepth(), gl.report.reproducibility(), gl.report.secondaries(), gl.report.sexlinked()

Examples

```
out <- gl.report.taglength(testset.gl)
```

`gl.run.structure` *Runs a STRUCTURE analysis using a genlight object*

Description

This function takes a genlight object and runs a STRUCTURE analysis based on functions from `strataG`

Usage

```
gl.run.structure(
  x,
  ...,
  exec = ".",
  plot.out = TRUE,
  plot_theme = theme_dartR(),
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object containing the SNP data [required].
`...`	Parameters to specify the STRUCTURE run (check `structureRun` within strataG. for more details). Parameters are passed to the `structureRun` function. For example you need to set the k.range and the type of model you would like to run (noadmix, locprior) etc. If those parameter names do not tell you anything, please make sure you familiarize with the STRUCTURE program (Pritchard 2000).
`exec`	Full path and name+extension where the structure executable is located. E.g. `'c:/structure/structure.exe'` under Windows. For Mac and Linux it might be something like `'./structure/structure'` if the executable is in a subfolder 'structure' in your home directory [default working directory "."].
`plot.out`	Create an Evanno plot once finished. Be aware k.range needs to be at least three different k steps [default TRUE].
`plot_theme`	Theme for the plot. See details for options [default theme_dartR()].
`save2tmp`	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
`verbose`	Set verbosity for this function (though structure output cannot be switched off currently) [default NULL]

Details

The function is basically a convenient wrapper around the beautiful strataG function `structureRun` (Archer et al. 2016). For a detailed description please refer to this package (see references below). To make use of this function you need to download STRUCTURE for you system (**non GUI version**) from here STRUCTURE.

Value

An sr object (structure.result list output). Each list entry is a single structurerun output (there are k.range * num.k.rep number of runs). For example the summary output of the first run can be accessed via `sr[[1]]$summary` or the q-matrix of the third run via `sr[[3]]$q.mat`. To conveniently summarise the outputs across runs (clumpp) you need to run gl.plot.structure on the returned sr object. For Evanno plots run gl.evanno on your sr object.

Author(s)

Bernd Gruber (Post to https://groups.google.com/d/forum/dartr)

References

- Pritchard, J.K., Stephens, M., Donnelly, P. (2000) Inference of population structure using multilocus genotype data. Genetics 155, 945-959.
- Archer, F. I., Adams, P. E. and Schneiders, B. B. (2016) strataG: An R package for manipulating, summarizing and analysing population genetic data. Mol Ecol Resour. doi:10.1111/1755-0998.12559

Examples

```
## Not run:
#bc <- bandicoot.gl[,1:100]
#sr <- gl.run.structure(bc, k.range = 2:5, num.k.rep = 3,
# exec = './structure.exe')
#ev <- gl.evanno(sr)
#ev
#qmat <- gl.plot.structure(sr, k=3, CLUMPP='d:/structure/')
#head(qmat)
#gl.map.structure(qmat, bc, scalex=1, scaley=0.5)

## End(Not run)
```

---

gl.save *Saves an object in compressed binary format for later rapid retrieval*

Description

This is a wrapper for saveRDS().

The script saves the object in binary form to the current workspace and returns the input gl object.

Usage

```
gl.save(x, file, verbose = NULL)
```

Arguments

x	Name of the genlight object containing SNP genotypes [required].
file	Name of the file to receive the binary version of the object [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].

Value

The input object

Author(s)

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

See Also

gl.load

Examples

```
gl.save(testset.gl,file.path(tempdir(),'testset.rds'))
```

---

gl.select.colors *Selects colors from one of several palettes and output as a vector*

Description

This script draws upon a number of specified color libraries to extract a vector of colors for plotting, where the script that follows has a color parameter expecting a vector of colors.

Usage

```
gl.select.colors(
  x = NULL,
  library = NULL,
  palette = NULL,
  ncolors = NULL,
  select = NULL,
  verbose = NULL
)
```

Arguments

x	Optionally, provide a gl object from which to determine the number of populations [default NULL].
library	Name of the color library to be used [default scales::hue_pl].
palette	Name of the color palette to be pulled from the specified library [default is library specific] .
ncolors	number of colors to be displayed and returned [default 9].
select	select the colors to retain in the output vector [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 available color libraries and their palettes include:

- library 'brewer' and the palettes available can be listed by RColorBrewer::display.brewer.all() and RColorBrewer::brewer.pal.info.
- library 'gr.palette' and the palettes available can be listed by grDevices::palette.pals()
- library 'r.hcl' and the palettes available can be listed by grDevices::hcl.pals()
- library 'baseR' and the palettes available are: 'rainbow','heat', 'topo.colors','terrain.colors','cm.colors'.

If the nominated palette is not specified, all the palettes will be listed a nd a default palette will then be chosen.

The color palette will be displayed in the graphics window for the requested number of colors (or 9 if not specified),and the vector of colors returned for later use.

The select parameter can be used to select colors from the specified ncolors. For example, select=c(1,1,3) will select color 1, 1 again and 3 to retain in the final vector. This can be useful for fine-tuning color selection, and matching colors and shapes.

Value

A vector with the required number of colors

Author(s)

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

See Also

gl.select.shapes

Other Exploration/visualisation functions: gl.pcoa.plot(), gl.select.shapes(), gl.smearplot()

Examples

```
# SET UP DATASET
gl <- testset.gl
levels(pop(gl))<-c(rep('Coast',5),rep('Cooper',3),rep('Coast',5),
rep('MDB',8),rep('Coast',7),'Em.subglobosa','Em.victoriae')
# EXAMPLES -- SIMPLE
colors <- gl.select.colors()
colors <- gl.select.colors(library='brewer',palette='Spectral',ncolors=6)
colors <- gl.select.colors(library='baseR',palette='terrain.colors',ncolors=6)
colors <- gl.select.colors(library='baseR',palette='rainbow',ncolors=12)
colors <- gl.select.colors(library='gr.hcl',palette='RdBu',ncolors=12)
colors <- gl.select.colors(library='gr.palette',palette='Pastel 1',ncolors=6)
# EXAMPLES -- SELECTING colorS
colors <- gl.select.colors(library='baseR',palette='rainbow',ncolors=12,select=c(1,1,1,5,8))
# EXAMPLES -- CROSS-CHECKING WITH A GENLIGHT OBJECT
colors <- gl.select.colors(x=gl,library='baseR',palette='rainbow',ncolors=12,select=c(1,1,1,5,8))
```

---

gl.select.shapes *Selects shapes from the base R shape palette and outputs as a vector*

Description

This script draws upon the standard R shape palette to extract a vector of shapes for plotting, where the script that follows has a shape parameter expecting a vector of shapes.

Usage

```
gl.select.shapes(x = NULL, select = NULL, verbose = NULL)
```

Arguments

x	Optionally, provide a gl object from which to determine the number of populations [default NULL].
select	Select the shapes to retain in the output vector [default NULL, all shapes shown and returned].
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

By default the shape palette will be displayed in full in the graphics window from which shapes can be selected in a subsequent run, and the vector of shapes returned for later use.

The select parameter can be used to select shapes from the specified 26 shapes available (0-25). For example, select=c(1,1,3) will select shape 1, 1 again and 3 to retain in the final vector. This can be useful for fine-tuning shape selection, and matching colors and shapes.

Value

A vector with the required number of shapes

Author(s)

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

See Also

gl.select.colors

Other Exploration/visualisation functions: gl.pcoa.plot(), gl.select.colors(), gl.smearplot()

Examples

```
# SET UP DATASET
gl <- testset.gl
levels(pop(gl))<-c(rep('Coast',5),rep('Cooper',3),rep('Coast',5),
rep('MDB',8),rep('Coast',7),'Em.subglobosa','Em.victoriae')
# EXAMPLES
shapes <- gl.select.shapes() # Select and display available shapes
# Select and display a restricted set of shapes
shapes <- gl.select.shapes(select=c(1,1,1,5,8))
 # Select set of shapes and check with no. of pops.
shapes <- gl.select.shapes(x=gl,select=c(1,1,1,5,8))
```

---

gl.set.verbosity *Sets the default verbosity level*

Description

dartR functions have a verbosity parameter that sets the level of reporting during the execution of the function. The verbosity level, set by parameter 'verbose' can be one of verbose 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report. The default value for verbosity is stored in the r environment. This script sets the default value.

Usage

```
gl.set.verbosity(value = 2)
```

Arguments

value	Set the default verbosity to be this value: 0, silent only fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2]

Value

verbosity value [set for all functions]

Author(s)

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

Examples

```
gl <- gl.set.verbosity(value=2)
```

---

gl.sfs *Creates a site frequency spectrum based on a dartR or genlight object*

Description

Creates a site frequency spectrum based on a dartR or genlight object

Usage

```
gl.sfs(
  x,
  minbinsize = 0,
  folded = TRUE,
  singlepop = FALSE,
  plot.out = TRUE,
  verbose = NULL
)
```

Arguments

x	dartR/genlight object
minbinsize	remove bins from the left of the sfs. For example to remove singletons (alleles only occurring once among all individuals) set minbinsize to 2. If set to zero, also monomorphic (d0) loci are returned.
folded	if set to TRUE (default) a folded sfs (minor allele frequency sfs) is returned. If set to FALSE then an unfolded (derived allele frequency sfs) is returned. It is assumed that 0 is homozygote for the reference and 2 is homozygote for the derived allele. So you need to make sure your coding is correct.
singlepop	switch to force to create a one-dimensional sfs, even though the genlight/dartR object contains more than one population
plot.out	Specify if plot is to be produced [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, unless specified using gl.set.verbosity].

Value

returns a site frequency spectrum, either a one dimensional vector (only a single population in the dartR/genlight object or singlepop=TRUE) or an n-dimensional array (n is the number of populations in the genlight/dartR object). If the dartR/genlight object consists of several populations the multidimensional site frequency spectrum for each population is returned [=a multidimensional site frequency spectrum]. Be aware the multidimensional spectrum works only for a limited number of population and individuals [if too high the table command used internally will through an error as the number of populations and individuals (and therefore dimensions) are too large]. To get a single sfs for a genlight/dartR object with multiple populations, you need to set singlepop to TRUE. The returned sfs can be used to analyse demographics, e.g. using fastsimcoal2.

Author(s)

Custodian: Bernd Gruber & Carlo Pacioni (Post to https://groups.google.com/d/forum/dartr)

References

Excoffier L., Dupanloup I., Huerta-Sánchez E., Sousa V. C. and Foll M. (2013) Robust demographic inference from genomic and SNP data. PLoS genetics 9(10)

Examples

```
gl.sfs(bandicoot.gl, singlepop=TRUE)
gl.sfs(possums.gl[c(1:5,31:33),], minbinsize=1)
```

---

gl.sim.create_dispersal

Creates a dispersal file as input for the function gl.sim.WF.run

---

Description

This function writes a csv file called "dispersal_table.csv" which contains the dispersal variables for each pair of populations to be used as input for the function gl.sim.WF.run.

The values of the variables can be modified using the columns "transfer_each_gen" and "number_transfers" of this file.

See documentation and tutorial for a complete description of the simulations. These documents can be accessed by typing in the R console: browseVignettes(package="dartR")

Usage

```
gl.sim.create_dispersal(
  number_pops,
  dispersal_type = "all_connected",
  number_transfers = 1,
  transfer_each_gen = 1,
  outpath = tempdir(),
```

```
  outfile = "dispersal_table.csv",
  verbose = NULL
)
```

Arguments

number_pops	Number of populations [required].
dispersal_type	One of: "all_connected", "circle" or "line" [default "all_connected"].
number_transfers	Number of dispersing individuals. This value can be . modified by hand after the file has been created [default 1].
transfer_each_gen	Interval of number of generations in which dispersal occur. This value can be modified by hand after the file has been created [default 1].
outpath	Path where to save the output file. Use outpath=getwd() or outpath='.' when calling this function to direct output files to your working directory [default tempdir(), mandated by CRAN].
outfile	File name of the output file [default 'dispersal_table.csv'].
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].

Value

A csv file containing the dispersal variables for each pair of populations to be used as input for the function gl.sim.WF.run.

Author(s)

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

See Also

gl.sim.WF.run

Other simulation functions: gl.sim.WF.run(), gl.sim.WF.table()

Examples

```
gl.sim.create_dispersal(number_pops=10)
```

`gl.sim.emigration` *Simulates emigration between populations*

Description

A function that allows to exchange individuals of populations within a genlight object (=simulate emigration between populations).

Usage

```
gl.sim.emigration(x, perc.mig = NULL, emi.m = NULL, emi.table = NULL)
```

Arguments

`x`	A genlight or list of genlight objects [required].
`perc.mig`	Percentage of individuals that migrate (emigrates = nInd times perc.mig) [default NULL].
`emi.m`	Probabilistic emigration matrix (emigrate from=column to=row) [default NULL]
`emi.table`	If presented emi.m matrix is ignored. Deterministic emigration as specified in the matrix (a square matrix of dimension of the number of populations). e.g. an entry in the 'emi.table[2,1]<- 5' means that five individuals emigrate from population 1 to population 2 (from=columns and to=row) [default NULL].

Details

There are two ways to specify emigration. If an emi.table is provided (a square matrix of dimension of the populations that specifies the emigration from column x to row y), then emigration is deterministic in terms of numbers of individuals as specified in the table. If perc.mig and emi.m are provided, then emigration is probabilistic. The number of emigrants is determined by the population size times the perc.mig and then the population where to migrate to is taken from the relative probability in the columns of the emi.m table.

Be aware if the diagonal is non zero then migration can occur into the same patch. So most often you want to set the diagonal of the emi.m matrix to zero. Which individuals is moved is random, but the order is in the order of populations. It is possible that an individual moves twice within an emigration call(as there is no check, so an individual moved from population 1 to 2 can move again from population 2 to 3).

Value

A list or a single [depends on the input] genlight object, where emigration between population has happened

Author(s)

Custodian: Bernd Gruber (Post to https://groups.google.com/d/forum/dartr)

Examples

```
x <- possums.gl
#one individual moves from every population to
#every other population
emi.tab <- matrix(1, nrow=nPop(x), ncol=nPop(x))
diag(emi.tab)<- 0
np <- gl.sim.emigration(x, emi.table=emi.tab)
np
```

`gl.sim.ind` *Simulates individuals based on the allele frequencies provided via a genlight object.*

Description

This function simulates individuals based on the allele frequencies of a genlight object. The output is a genlight object with the same number of loci as the input genlight object.

Usage

```
gl.sim.ind(x, n = 50, popname = NULL)
```

Arguments

`x`	Name of the genlight object containing the SNP data [required].
`n`	Number of individuals that should be simulated [default 50].
`popname`	A population name for the simulated individuals [default NULL].

Details

The function can be used to simulate populations for sampling designs or for power analysis. Check the example below where the effect of drift is explored, by simply simulating several generation a genlight object and putting in the allele frequencies of the previous generation. The beauty of the function is, that it is lightning fast. Be aware this is a simulation and to avoid lengthy error checking the function crashes if there are loci that have just NAs. If such a case can occur during your simulation, those loci need to be removed, before the function is called.

Value

A genlight object with n individuals.

Author(s)

Bernd Gruber (bernd.gruber@canberra.edu.au)

Examples

```
glsim <- gl.sim.ind(testset.gl, n=10, popname='sims')
glsim
###Simulate drift over 10 generation
# assuming a bottleneck of only 10 individuals
# [ignoring effect of mating and mutation]
# Simulate 20 individuals with no structure and 50 SNP loci
founder <- glSim(n.ind = 20, n.snp.nonstruc = 50, ploidy=2)
#number of fixed loci in the first generation

res <- sum(colMeans(as.matrix(founder), na.rm=TRUE) %%2 ==0)
simgl <- founder
#49 generations of only 10 individuals
for (i in 2:50)
{
   simgl <- gl.sim.ind(simgl, n=10, popname='sims')
   res[i]<- sum(colMeans(as.matrix(simgl), na.rm=TRUE) %%2 ==0)
}
plot(1:50, res, type='b', xlab='generation', ylab='# fixed loci')
```

---

gl.sim.mutate *Simulates mutations within a genlight object*

Description

This script is intended to be used within the simulation framework of dartR. It adds the ability to add a constant mutation rate across all loci. Only works currently for biallelic data sets (SNPs). Mutation rate is checking for all alleles position and mutations at loci with missing values are ignored and in principle 'double mutations' at the same loci can occur, but should be rare.

Usage

```
gl.sim.mutate(x, mut.rate = 1e-06)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
mut.rate	Constant mutation rate over nInd*nLoc*2 possible locations [default 1e-6]

Value

Returns a genlight object with the applied mutations

Author(s)

Bernd Gruber (Post to https://groups.google.com/d/forum/dartr)

Examples

```
b2 <- gl.sim.mutate(bandicoot.gl,mut.rate=1e-4 )
#check the mutations that have occurred
table(as.matrix(bandicoot.gl), as.matrix(b2))
```

`gl.sim.offspring` *Simulates a specified number of offspring based on alleles provided by potential father(s) and mother(s)*

Description

This takes a population (or a single individual) of fathers (provided as a genlight object) and mother(s) and simulates offspring based on 'random' mating. It can be used to simulate population dynamics and check the effect of those dynamics and allele frequencies, number of alleles. Another application is to simulate relatedness of siblings and compare it to actual relatedness found in the population to determine kinship.

Usage

```
gl.sim.offspring(fathers, mothers, noffpermother, sexratio = 0.5)
```

Arguments

`fathers`	Genlight object of potential fathers [required].
`mothers`	Genlight object of potential mothers simulated [required].
`noffpermother`	Number of offspring per mother [required].
`sexratio`	The sex ratio of simulated offspring (females / females +males, 1 equals 100 percent females) [default 0.5.].

Value

A genlight object with n individuals.

Author(s)

Bernd Gruber (Post to https://groups.google.com/d/forum/dartr)

Examples

```
#Simulate 10 potential fathers
gl.fathers <- glSim(10, 20, ploidy=2)
#Simulate 10 potential mothers
gl.mothers <- glSim(10, 20, ploidy=2)
gl.sim.offspring(gl.fathers, gl.mothers, 2, sexratio=0.5)
```

`gl.sim.WF.run` *Runs Wright-Fisher simulations*

Description

This function simulates populations made up of diploid organisms that reproduce in non-overlapping generations. Each individual has a pair of homologous chromosomes that contains interspersed selected and neutral loci. For the initial generation, the genotype for each individual's chromosomes is randomly drawn from distributions at linkage equilibrium and in Hardy-Weinberg equilibrium.

See documentation and tutorial for a complete description of the simulations. These documents can be accessed at http://georges.biomatix.org/dartR

Take into account that the simulations will take a little bit longer the first time you use the function gl.sim.WF.run() because C++ functions must be compiled.

Usage

```
gl.sim.WF.run(
  file_var,
  ref_table,
  x = NULL,
  file_dispersal = NULL,
  number_iterations = 1,
  every_gen = 10,
  sample_percent = 50,
  store_phase1 = FALSE,
  interactive_vars = TRUE,
  seed = NULL,
  verbose = NULL,
  ...
)
```

Arguments

`file_var`	Path of the variables file 'sim_variables.csv' (see details) [required if interactive_vars = FALSE].
`ref_table`	Reference table created by the function `gl.sim.WF.table` [required].
`x`	Name of the genlight object containing the SNP data to extract values for some simulation variables (see details) [default NULL].
`file_dispersal`	Path of the file with the dispersal table created with the function `gl.sim.create_dispersal` [default NULL].
`number_iterations`	Number of iterations of the simulations [default 1].
`every_gen`	Generation interval at which simulations should be stored in a genlight object [default 10].

sample_percent	Percentage of individuals, from the total population, to sample and save in the genlight object every generation [default 50].
store_phase1	Whether to store simulations of phase 1 in genlight objects [default FALSE].
interactive_vars	Run a shiny app to input interactively the values of simulations variables [default TRUE].
seed	Set the seed for the simulations [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, unless specified using gl.set.verbosity].
...	Any variable and its value can be added separately within the function, will be changed over the input value supplied by the csv file. See tutorial.

Details

Values for simulation variables can be submitted into the function interactively through a shiny app if interactive_vars = TRUE. Optionally, if interactive_vars = FALSE, values for variables can be submitted by using the csv file 'sim_variables.csv' which can be found by typing in the R console: system.file('extdata', 'sim_variables.csv', package ='dartR').

The values of the variables can be modified using the third column ("value") of this file.

The output of the simulations can be analysed seemingly with other dartR functions.

If a genlight object is used as input for some of the simulation variables, this function access the information stored in the slots x$position and x$chromosome.

To show further information of the variables in interactive mode, it might be necessary to call first: 'library(shinyBS)' for the information to be displayed.

The main characteristics of the simulations are:

- Simulations can be parameterised with real-life genetic characteristics such as the number, location, allele frequency and the distribution of fitness effects (selection coefficients and dominance) of loci under selection.
- Simulations can recreate specific life histories and demographics, such as source populations, dispersal rate, number of generations, founder individuals, effective population size and census population size.
- Each allele in each individual is an agent (i.e., each allele is explicitly simulated).
- Each locus can be customisable regarding its allele frequencies, selection coefficients, and dominance.
- The number of loci, individuals, and populations to be simulated is only limited by computing resources.
- Recombination is accurately modeled, and it is possible to use real recombination maps as input.
- The ratio between effective population size and census population size can be easily controlled.
- The output of the simulations are genlight objects for each generation or a subset of generations.
- Genlight objects can be used as input for some simulation variables.

Value

Returns genlight objects with simulated data.

Author(s)

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

See Also

gl.sim.WF.table

Other simulation functions: gl.sim.WF.table(), gl.sim.create_dispersal()

Examples

```
## Not run:
ref_table <- gl.sim.WF.table(file_var=system.file('extdata',
'ref_variables.csv', package = 'dartR'),interactive_vars = FALSE)
res_sim <- gl.sim.WF.run(file_var = system.file('extdata',
'sim_variables.csv', package ='dartR'),ref_table=ref_table,
interactive_vars = FALSE)

## End(Not run)
```

---

gl.sim.WF.table *Creates the reference table for running gl.sim.WF.run*

Description

This function creates a reference table to be used as input for the function gl.sim.WF.run. The created table has eight columns with the following information for each locus to be simulated:

- q - initial frequency.
- h - dominance coefficient.
- s - selection coefficient.
- c - recombination rate.
- loc_bp - chromosome location in base pairs.
- loc_cM - chromosome location in centiMorgans.
- chr_name - chromosome name.
- type - SNP type.

The reference table can be further modified as required.

See documentation and tutorial for a complete description of the simulations. These documents can be accessed at http://georges.biomatix.org/dartR

Usage

```
gl.sim.WF.table(
  file_var,
  x = NULL,
  file_targets_sel = NULL,
  file_r_map = NULL,
  interactive_vars = TRUE,
  seed = NULL,
  verbose = NULL,
  ...
)
```

Arguments

file_var	Path of the variables file 'ref_variables.csv' (see details) [required if interactive_vars = FALSE].
x	Name of the genlight object containing the SNP data to extract values for some simulation variables (see details) [default NULL].
file_targets_sel	Path of the file with the targets for selection (see details) [default NULL].
file_r_map	Path of the file with the recombination map (see details) [default NULL].
interactive_vars	Run a shiny app to input interactively the values of simulation variables [default TRUE].
seed	Set the seed for the simulations [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, unless specified using gl.set.verbosity].
...	Any variable and its value can be added separately within the function, will be changed over the input value supplied by the csv file. See tutorial.

Details

Values for the variables to create the reference table can be submitted into the function interactively through a Shiny app if interactive_vars = TRUE. Optionally, if interactive_vars = FALSE, values for variables can be submitted by using the csv file 'ref_variables.csv' which can be found by typing in the R console: system.file('extdata', 'ref_variables.csv', package ='dartR').

The values of the variables can be modified using the third column ("value") of this file.

If a genlight object is used as input for some of the simulation variables, this function access the information stored in the slots x$position and x$chromosome.

Examples of the format required for the recombination map file and the targets for selection file can be found by typing in the R console:

- system.file('extdata', 'fly_recom_map.csv', package ='dartR')
- system.file('extdata', 'fly_targets_of_selection.csv', package ='dartR')

To show further information of the variables in interactive mode, it might be necessary to call first: 'library(shinyBS)' for the information to be displayed.

Value

Returns a list with the reference table used as input for the function gl.sim.WF.run and a table with the values variables used to create the reference table.

Author(s)

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

See Also

gl.sim.WF.run

Other simulation functions: gl.sim.WF.run(), gl.sim.create_dispersal()

Examples

```
ref_table <- gl.sim.WF.table(file_var=system.file('extdata',
'ref_variables.csv', package = 'dartR'),interactive_vars = FALSE)
## Not run:
#uncomment to run
res_sim <- gl.sim.WF.run(file_var = system.file('extdata',
'sim_variables.csv', package ='dartR'),ref_table=ref_table,
interactive_vars = FALSE)

## End(Not run)
```

---

gl.smearplot *Smear plot of SNP or presence/absence (SilicoDArT) data*

Description

Each locus is color coded for scores of 0, 1, 2 and NA for SNP data and 0, 1 and NA for presence/absence (SilicoDArT) data. Individual labels can be added and individuals can be grouped by population.

Plot may become cluttered if ind_labels If there are too many individuals, it is best to use ind_labels_size = 0.

Usage

```
gl.smearplot(
  x,
  ind_labels = FALSE,
  group_pop = FALSE,
  ind_labels_size = 10,
  plot_colors = colorRampPalette(c("royalblue3", "firebrick1"))(3),
  posi = "bottom",
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP or presence/absence (SilicoDArT) data [required].
ind_labels	If TRUE, individuals are labelled with indNames(x) [default FALSE].
group_pop	If ind_labels is TRUE, group by population [default TRUE].
ind_labels_size	Size of the individual labels [default 10].
plot_colors	Vector with four color names for homozygotes for the reference allele, heterozygotes, homozygotes for the alternative allele and for missing values (NA), e.g. four_colours [default NULL].
posi	Position of the legend: "left", "top", "right", "bottom" or 'none' [default = 'bottom'].
save2tmp	If TRUE, saves plot to the session temporary directory (tempdir) [default FALSE].
verbose	Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default NULL].

Value

Returns unaltered genlight object

Author(s)

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

See Also

Other Exploration/visualisation functions: gl.pcoa.plot(), gl.select.colors(), gl.select.shapes()

Examples

```
gl.smearplot(testset.gl,ind_labels=FALSE)
gl.smearplot(testset.gs[1:10,],ind_labels=TRUE)
```

---

gl.spatial.autoCorr *Spatial autocorrelation following Smouse and Peakall 1999*

Description

Global spatial autocorrelation is a multivariate approach combining all loci into a single analysis. The autocorrelation coefficient "r" is calculated for each pair of individuals in each specified distance class. For more information see Smouse and Peakall 1999, Peakall et al. 2003 and Smouse et al. 2008.

Usage

```
gl.spatial.autoCorr(
  x = NULL,
  Dgeo = NULL,
  Dgen = NULL,
  coordinates = "latlon",
  Dgen_method = "Euclidean",
  Dgeo_trans = "Dgeo",
  Dgen_trans = "Dgen",
  bins = 5,
  reps = 100,
  plot.pops.together = FALSE,
  permutation = TRUE,
  bootstrap = TRUE,
  plot_theme = NULL,
  plot_colors_pop = NULL,
  CI_color = "red",
  plot.out = TRUE,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	Genlight object [default NULL].
Dgeo	Geographic distance matrix if no genlight object is provided. This is typically an Euclidean distance but it can be any meaningful (geographical) distance metrics [default NULL].
Dgen	Genetic distance matrix if no genlight object is provided [default NULL].
coordinates	Can be either 'latlon', 'xy' or a two column data.frame with column names 'lat','lon', 'x', 'y') Coordinates are provided via `gl@other$latlon` ['latlon'] or via `gl@other$xy` ['xy']. If latlon data will be projected to meters using Mercator system [google maps] or if xy then distance is directly calculated on the coordinates [default .
Dgen_method	Method to calculate genetic distances. See details [default "Euclidean"].
Dgeo_trans	Transformation to be used on the geographic distances. See Dgen_trans [default "Dgeo"].
Dgen_trans	You can provide a formula to transform the genetic distance. The transformation can be applied as a formula using Dgen as the variable to be transformed. For example: `Dgen_trans = 'Dgen/(1-Dgen)'`. Any valid R expression can be used here [default 'Dgen', which is the identity function.]
bins	The number of bins for the distance classes (i.e. `length(bins) == 1`) or a vectors with the break points. See details [default 5].
reps	The number to be used for permutation and bootstrap analyses [default 100].

plot.pops.together
: Plot all the populations in one plot. Confidence intervals from permutations are not shown [default FALSE].

permutation
: Whether permutation calculations for the null hypothesis of no spatial structure should be carried out [default TRUE].

bootstrap
: Whether bootstrap calculations to compute the 95% confidence intervals around r should be carried out [default TRUE].

plot_theme
: Theme for the plot. See details [default NULL].

plot_colors_pop
: A color palette for populations or a list with as many colors as there are populations in the dataset [default NULL].

CI_color
: Color for the shade of the 95% confidence intervals around the r estimates [default "red"].

plot.out
: Specify if plot is to be produced [default TRUE].

save2tmp
: If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].

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

This function executes a modified version of `spautocorr` from the package `PopGenReport`. Differently from `PopGenReport`, this function also computes the 95% confidence intervals around the r via bootstraps, the 95 null hypothesis of no spatial structure and the one-tail test via permutation, and the correction factor described by Peakall et al 2003.

The input can be i) a genlight object (which has to have the latlon slot populated), ii) a pair of `Dgeo` and `Dgen`, which have to be either `matrix` or `dist` objects, or iii) a `list` of the `matrix` or `dist` objects if the analysis needs to be carried out for multiple populations (in this case, all the elements of the `list` have to be of the same class (i.e. `matrix` or `dist`) and the population order in the two lists has to be the same.

If the input is a genlight object, the function calculates the linear distance for `Dgeo` and the relevant `Dgen` matrix (see `Dgen_method`) for each population. When the method selected is a genetic similarity matrix (e.g. "simple" distance), the matrix is internally transformed with `1 - Dgen` so that positive values of autocorrelation coefficients indicates more related individuals similarly as implemented in GenAlEx. If the user provide the distance matrices, care must be taken in interpreting the results because similarity matrix will generate negative values for closely related individuals.

If `max(Dgeo)>1000` (e.g. the geographic distances are in thousands of metres), values are divided by 1000 (in the example before these would then become km) to facilitate readability of the plots.

If `bins` is of length = 1 it is interpreted as the number of (even) bins to use. In this case the starting point is always the minimum value in the distance matrix, and the last is the maximum. If it is a numeric vector of length>1, it is interpreted as the breaking points. In this case, the first has to be the lowest value, and the last has to be the highest. There are no internal checks for this and it is user responsibility to ensure that distance classes are properly set up. If that is not the case, data that fall outside the range provided will be dropped. The number of bins will be `length(bins) - 1`.

The permutation constructs the 95% confidence intervals around the null hypothesis of no spatial structure (this is a two-tail test). The same data are also used to calculate the probability of the one-tail test (See references below for details).

Bootstrap calculations are skipped and NA is returned when the number of possible combinations given the sample size of any given distance class is < reps.

Methods available to calculate genetic distances for SNP data:

- "propShared" using the function gl.propShared.
- "grm" using the function gl.grm.
- "Euclidean" using the function gl.dist.ind.
- "Simple" using the function gl.dist.ind.
- "Absolute" using the function gl.dist.ind.
- "Manhattan" using the function gl.dist.ind.

Methods available to calculate genetic distances for SilicoDArT data:

- "Euclidean" using the function gl.dist.ind.
- "Simple" using the function gl.dist.ind.
- "Jaccard" using the function gl.dist.ind.
- "Bray-Curtis" using the function gl.dist.ind.

Plots and table are saved to the temporal directory (tempdir) and can be accessed with the function gl.print.reports and listed with the function gl.list.reports. Note that they can be accessed only in the current R session because tempdir is cleared each time that the R session is closed.

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

Returns a data frame with the following columns:

1. Bin The distance classes
2. N The number of pairwise comparisons within each distance class
3. r.uc The uncorrected autocorrelation coefficient
4. Correction the correction
5. r The corrected autocorrelation coefficient
6. L.r The corrected autocorrelation coefficient lower limit (if bootstap = TRUE)
7. U.r The corrected autocorrelation coefficient upper limit (if bootstap = TRUE)
8. L.r.null.uc The uncorrected lower limit for the null hypothesis of no spatial autocorrelation (if permutation = TRUE)
9. U.r.null.uc The uncorrected upper limit for the null hypothesis of no spatial autocorrelation (if permutation = TRUE)

10. L.r.null The corrected lower limit for the null hypothesis of no spatial autocorrelation (if `permutation = TRUE`)
11. U.r.null The corrected upper limit for the null hypothesis of no spatial autocorrelation (if `permutation = TRUE`)
12. p.one.tail The p value of the one tail statistical test

Author(s)

Carlo Pacioni, Bernd Gruber & Luis Mijangos (Post to https://groups.google.com/d/forum/dartr)

References

- Smouse PE, Peakall R. 1999. Spatial autocorrelation analysis of individual multiallele and multilocus genetic structure. Heredity 82: 561-573.
- Double, MC, et al. 2005. Dispersal, philopatry and infidelity: dissecting local genetic structure in superb fairy-wrens (Malurus cyaneus). Evolution 59, 625-635.
- Peakall, R, et al. 2003. Spatial autocorrelation analysis offers new insights into gene flow in the Australian bush rat, Rattus fuscipes. Evolution 57, 1182-1195.
- Smouse, PE, et al. 2008. A heterogeneity test for fine-scale genetic structure. Molecular Ecology 17, 3389-3400.
- Gonzales, E, et al. 2010. The impact of landscape disturbance on spatial genetic structure in the Guanacaste tree, Enterolobium cyclocarpum (Fabaceae). Journal of Heredity 101, 133-143.
- Beck, N, et al. 2008. Social constraint and an absence of sex-biased dispersal drive fine-scale genetic structure in white-winged choughs. Molecular Ecology 17, 4346-4358.

Examples

```
require("dartR.data")
res <- gl.spatial.autoCorr(platypus.gl, bins=seq(0,10000,2000))
# using one population, showing sample size
test <- gl.keep.pop(platypus.gl,pop.list = "TENTERFIELD")
res <- gl.spatial.autoCorr(test, bins=seq(0,10000,2000),CI_color = "green")

test <- gl.keep.pop(platypus.gl,pop.list = "TENTERFIELD")
res <- gl.spatial.autoCorr(test, bins=seq(0,10000,2000),CI_color = "green")
```

---

gl.subsample.loci *Subsamples n loci from a genlight object and return it as a genlight object*

Description

This is a support script, to subsample a genlight {adegenet} object based on loci. Two methods are used to subsample, random and based on information content.

Usage

```
gl.subsample.loci(x, n, method = "random", mono.rm = FALSE, verbose = NULL)
```

Arguments

x	Name of the genlight object containing the SNP or presence/absence (SilicoDArT) data [required].
n	Number of loci to include in the subsample [required].
method	Method: 'random', in which case the loci are sampled at random; or 'pic', in which case the top n loci ranked on information content are chosen. Information content is stored in AvgPIC in the case of SNP data and in PIC in the the case of presence/absence (SilicoDArT) data [default 'random'].
mono.rm	Delete monomorphic loci before sampling [default FALSE].
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 with n loci

Author(s)

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

Examples

```
# SNP data
  gl2 <- gl.subsample.loci(testset.gl, n=200, method='pic')
# Tag P/A data
  gl2 <- gl.subsample.loci(testset.gl, n=100, method='random')
```

---

gl.test.heterozygosity

Tests the difference in heterozygosity between populations taken pairwise

---

Description

Calculates the expected heterozygosities for each population in a genlight object, and uses randomization to test the statistical significance of differences in heterozygosity between populations taken pairwise.

Usage

```
gl.test.heterozygosity(
  x,
  nreps = 100,
  alpha1 = 0.05,
  alpha2 = 0.01,
  plot.out = TRUE,
  max_plots = 6,
  plot_theme = theme_dartR(),
  plot_colors = two_colors,
  save2tmp = FALSE,
  verbose = NULL
)
```

Arguments

x	A genlight object containing the SNP genotypes [required].
nreps	Number of replications of the re-randomization [default 1,000].
alpha1	First significance level for comparison with diff=0 on plot [default 0.05].
alpha2	Second significance level for comparison with diff=0 on plot [default 0.01].
plot.out	If TRUE, plots a sampling distribution of the differences for each comparison [default TRUE].
max_plots	Maximum number of plots to print per page [default 6].
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 two_colors].
save2tmp	If TRUE, saves any ggplots and listings to the session temporary directory (tempdir) [default FALSE].
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

Function's output

If plot.out = TRUE, plots are created showing the sampling distribution for the difference between each pair of heterozygosities, marked with the critical limits alpha1 and alpha2, the observed heterozygosity, and the zero value (if in range).

Plots and table are saved to the temporal directory (tempdir) and can be accessed with the function gl.print.reports and listed with the function gl.list.reports. Note that they can be accessed only in the current R session because tempdir is cleared each time that the R session is closed.

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 dataframe containing population labels, heterozygosities and sample sizes

Author(s)

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

Examples

```
out <- gl.test.heterozygosity(platypus.gl, nreps=1, verbose=3, plot.out=TRUE)
```

---

gl.tree.nj *Outputs an nj tree to summarize genetic similarity among populations*

Description

This function is a wrapper for the nj{ape} function applied to Euclidian distances calculated from the genlight object.

Usage

```
gl.tree.nj(
  x,
  type = "phylogram",
  outgroup = NULL,
  labelsize = 0.7,
  treefile = NULL,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
type	Type of dendrogram "phylogram"|"cladogram"|"fan"|"unrooted" [default "phylogram"].
outgroup	Vector containing the population names that are the outgroups [deefault NULL].
labelsize	Size of the labels as a proportion of the graphics default [default 0.7].
treefile	Name of the file for the tree topology using Newick format [default NULL].
verbose	Specify the level of verbosity: 0, silent, fatal errors only; 1, flag function begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2].

Value

A tree file of class phylo.

Author(s)

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

Examples

```
# SNP data
  gl.tree.nj(testset.gl,type='fan')
# Tag P/A data
  gl.tree.nj(testset.gs,type='fan')

  res <- gl.tree.nj(platypus.gl)
```

---

gl.write.csv *Writes out data from a genlight object to csv file*

Description

This script writes to file the SNP genotypes with specimens as entities (columns) and loci as attributes (rows). Each row has associated locus metadata. Each column, with header of specimen id, has population in the first row.

The data coding differs from the DArT 1row format in that 0 = reference homozygous, 2 = alternate homozygous, 1 = heterozygous, and NA = missing SNP assignment.

Usage

```
gl.write.csv(x, outfile = "outfile.csv", outpath = tempdir(), verbose = NULL)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
outfile	File name of the output file (including extension) [default "outfile.csv"].
outpath	Path where to save the output file [default tempdir(), mandated by CRAN]. Use outpath=getwd() or outpath='.' when calling this function to direct output files to your 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].

Value

Saves a genlight object to csv, returns NULL.

Author(s)

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

Examples

```
# SNP data
  gl.write.csv(testset.gl, outfile='SNP_1row.csv')
# Tag P/A data
  gl.write.csv(testset.gs, outfile='PA_1row.csv')
```

---

gl2bayescan *Converts a genlight object into a format suitable for input to Bayescan*

Description

The output text file contains the SNP data and relevant BAyescan command lines to guide input.

Usage

```
gl2bayescan(x, outfile = "bayescan.txt", outpath = tempdir(), verbose = NULL)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
outfile	File name of the output file (including extension) [default bayescan.txt].
outpath	Path where to save the output file [default tempdir(), mandated by CRAN]. Use outpath=getwd() or outpath='.' when calling this function to direct output files to your 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].

Author(s)

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

References

Foll M and OE Gaggiotti (2008) A genome scan method to identify selected loci appropriate for both dominant and codominant markers: A Bayesian perspective. Genetics 180: 977-993.

Examples

```
out <- gl2bayescan(testset.gl)
```

gl2bpp *Converts a genlight object into a format suitable for input to the BPP program*

Description

This function generates the sequence alignment file and the Imap file. The control file should produced by the user.

Usage

```
gl2bpp(
  x,
  method = 1,
  outfile = "output_bpp.txt",
  imap = "Imap.txt",
  outpath = tempdir(),
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
method	One of 1 | 2, see details [default = 1].
outfile	Name of the sequence alignment file ["output_bpp.txt"].
imap	Name of the Imap file ["Imap.txt"].
outpath	Path where to save the output file (set to tempdir by default)
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

If method = 1, heterozygous positions are replaced by standard ambiguity codes.

If method = 2, the heterozygous state is resolved by randomly assigning one or the other SNP variant to the individual.

Trimmed sequences for which the SNP has been trimmed out, rarely, by adapter mis-identity are deleted.

This function requires 'TrimmedSequence' to be among the locus metrics (`@other$loc.metrics`) and information of the type of alleles (slot loc.all e.g. 'G/A') and the position of the SNP in slot position of the "'genlight"' object (see testset.gl@position and testset.gl@loc.all for how to format these slots.)

It's important to keep in mind that analyses based on coalescent theory, like those done by the programme BPP, are meant to be used with sequence data. In this type of data, large chunks of DNA are sequenced, so when we find polymorphic sites along the sequence, we know they are all

on the same chromosome. This kind of data, in which we know which chromosome each allele comes from, is called "phased data." Most data from reduced representation genome-sequencing methods, like DArTseq, is unphased, which means that we don't know which chromosome each allele comes from. So, if we apply coalescence theory to data that is not phased, we will get biased results. As in Ellegren et al., one way to deal with this is to "haplodize" each genotype by randomly choosing one allele from heterozygous genotypes (2012) by using method = 2.

Be mindful that there is little information in the literature on the validity of this method.

Author(s)

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

References

- Ellegren, Hans, et al. "The genomic landscape of species divergence in Ficedula flycatchers." Nature 491.7426 (2012): 756-760.
- Flouri T., Jiao X., Rannala B., Yang Z. (2018) Species Tree Inference with BPP using Genomic Sequences and the Multispecies Coalescent. Molecular Biology and Evolution, 35(10):2585-2593. doi:10.1093/molbev/msy147

Examples

```
require(dartR.data)
test <- platypus.gl
test <- gl.filter.callrate(test,threshold = 1)
test <- gl.filter.monomorphs(test)
test <- gl.subsample.loci(test,n=25)
gl2bpp(x = test)
```

`gl2demerelate` *Creates a dataframe suitable for input to package {Demerelate} from a genlight {adegenet} object*

Description

Creates a dataframe suitable for input to package {Demerelate} from a genlight {adegenet} object

Usage

```
gl2demerelate(x, verbose = NULL)
```

Arguments

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

Value

A dataframe suitable as input to package {Demerelate}

Author(s)

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

Examples

```
df <- gl2demerelate(testset.gl)
```

`gl2eigenstrat` *Converts a genlight object into eigenstrat format*

Description

The output of this function are three files:

- genotype file: contains genotype data for each individual at each SNP with an extension 'eigenstratgeno.'
- snp file: contains information about each SNP with an extension 'snp.'
- indiv file: contains information about each individual with an extension 'ind.'

Usage

```
gl2eigenstrat(
  x,
  outfile = "gl_eigenstrat",
  outpath = tempdir(),
  snp_pos = 1,
  snp_chr = 1,
  pos_cM = 0,
  sex_code = "unknown",
  phen_value = "Case",
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object containing the SNP data [required].
`outfile`	File name of the output file [default 'gl_eigenstrat'].
`outpath`	Path where to save the output file [default tempdir(), mandated by CRAN]. Use outpath=getwd() or outpath='.' when calling this function to direct output files to your working directory.
`snp_pos`	Field name from the slot loc.metrics where the SNP position is stored [default 1].

snp_chr	Field name from the slot loc.metrics where the chromosome of each is stored [default 1].
pos_cM	A vector, with as many elements as there are loci, containing the SNP position in morgans or centimorgans [default 1].
sex_code	A vector, with as many elements as there are individuals, containing the sex code ('male', 'female', 'unknown') [default 'unknown'].
phen_value	A vector, with as many elements as there are individuals, containing the phenotype value ('Case', 'Control') [default 'Case'].
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

Eigenstrat only accepts chromosomes coded as numeric values, as follows: X chromosome is encoded as 23, Y is encoded as 24, mtDNA is encoded as 90, and XY is encoded as 91. SNPs with illegal chromosome values, such as 0, will be removed.

Author(s)

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

References

- Patterson, N., Price, A. L., & Reich, D. (2006). Population structure and eigenanalysis. PLoS genetics, 2(12), e190.
- Price, A. L., Patterson, N. J., Plenge, R. M., Weinblatt, M. E., Shadick, N. A., & Reich, D. (2006). Principal components analysis corrects for stratification in genome-wide association studies. Nature genetics, 38(8), 904-909.

Examples

```
require("dartR.data")
gl2eigenstrat(platypus.gl,snp_pos='ChromPos_Platypus_Chrom_NCBIv1',
snp_chr = 'Chrom_Platypus_Chrom_NCBIv1')
```

---

gl2fasta *Concatenates DArT trimmed sequences and outputs a FASTA file*

Description

Concatenated sequence tags are useful for phylogenetic methods where information on base frequencies and transition and transversion ratios are required (for example, Maximum Likelihood methods). Where relevant, heterozygous loci are resolved before concatenation by either assigning ambiguity codes or by random allele assignment.

Usage

```
gl2fasta(
  x,
  method = 1,
  outfile = "output.fasta",
  outpath = tempdir(),
  probar = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
method	One of 1 | 2 | 3 | 4. Type method=0 for a list of options [method=1].
outfile	Name of the output file (fasta format) ["output.fasta"].
outpath	Path where to save the output file (set to tempdir by default)
probar	If TRUE, a progress bar will be displayed for long loops [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

Four methods are employed:

Method 1 – heterozygous positions are replaced by the standard ambiguity codes. The resultant sequence fragments are concatenated across loci to generate a single combined sequence to be used in subsequent ML phylogenetic analyses.

Method 2 – the heterozygous state is resolved by randomly assigning one or the other SNP variant to the individual. The resultant sequence fragments are concatenated across loci to generate a single composite haplotype to be used in subsequent ML phylogenetic analyses.

Method 3 – heterozygous positions are replaced by the standard ambiguity codes. The resultant SNP bases are concatenated across loci to generate a single combined sequence to be used in subsequent MP phylogenetic analyses.

Method 4 – the heterozygous state is resolved by randomly assigning one or the other SNP variant to the individual. The resultant SNP bases are concatenated across loci to generate a single composite haplotype to be used in subsequent MP phylogenetic analyses.

Trimmed sequences for which the SNP has been trimmed out, rarely, by adapter mis-identity are deleted.

The script writes out the composite haplotypes for each individual as a fastA file. Requires 'TrimmedSequence' to be among the locus metrics (`@other$loc.metrics`) and information of the type of alleles (slot loc.all e.g. 'G/A') and the position of the SNP in slot position of the "'genlight"' object (see testset.gl@position and testset.gl@loc.all for how to format these slots.)

Value

A new gl object with all loci rendered homozygous.

Author(s)

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

Examples

```
gl <- gl.filter.reproducibility(testset.gl,t=1)
gl <- gl.filter.overshoot(gl,verbose=3)
gl <- gl.filter.callrate(testset.gl,t=.98)
gl <- gl.filter.monomorphs(gl)
gl2fasta(gl, method=1, outfile='test.fasta',verbose=3)

test <- gl.subsample.loci(platypus.gl,n=100)
gl2fasta(test)
```

---

gl2faststructure *Converts a genlight object into faststructure format (to run faststructure elsewhere)*

Description

Recodes in the quite specific faststructure format (e.g first six columns need to be there, but are ignored...check faststructure documentation (if you find any :-( ))

Usage

```
gl2faststructure(
  x,
  outfile = "gl.str",
  outpath = tempdir(),
  probar = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
outfile	File name of the output file (including extension) [default "gl.str"].
outpath	Path where to save the output file [default tempdir(), mandated by CRAN]. Use outpath=getwd() or outpath='.' when calling this function to direct output files to your working directory.
probar	Switch to show/hide progress bar [default FALSE].
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 script writes out the a file in faststructure format.

Author(s)

Bernd Gruber (Post to https://groups.google.com/d/forum/dartr)

`gl2gds` *Converts a genlight object into gds format*

Description

Package SNPRelate relies on a bit-level representation of a SNP dataset that competes with {adegenet} genlight objects and associated files. This function converts a genlight object to a gds format file.

Usage

```
gl2gds(
  x,
  outfile = "gl_gds.gds",
  outpath = tempdir(),
  snp_pos = "0",
  snp_chr = "0",
  chr_format = "character",
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object containing the SNP data [required].
`outfile`	File name of the output file (including extension) [default 'gl_gds.gds'].
`outpath`	Path where to save the output file [default tempdir(), mandated by CRAN]. Use outpath=getwd() or outpath='.' when calling this function to direct output files to your working directory.
`snp_pos`	Field name from the slot loc.metrics where the SNP position is stored [default '0'].
`snp_chr`	Field name from the slot loc.metrics where the chromosome of each is stored [default '0'].
`chr_format`	Whether chromosome information is stored as 'numeric' or as 'character', see details [default 'character'].
`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 orders the SNPS by chromosome and by position before converting to SNPRelate format, as required by this package.

The chromosome of each SNP can be a character or numeric, as described in the vignette of SNPRelate: 'snp.chromosome, an integer or character mapping for each chromosome. Integer: numeric values 1-26, mapped in order from 1-22, 23=X, 24=XY (the pseudoautosomal region), 25=Y, 26=M (the mitochondrial probes), and 0 for probes with unknown positions; it does not allow NA. Character: "X", "XY", "Y" and "M" can be used here, and a blank string indicating unknown position.'

When using some functions from package SNPRelate with datasets other than humans it might be necessary to use the option autosome.only=FALSE to avoid detecting chromosome coding. So, it is important to read the documentation of the function before using it.

The chromosome information for unmapped SNPS is coded as 0, as required by SNPRelate.

Remember to close the GDS file before working in a different GDS object with the function snpgdsClose (package SNPRelate).

Author(s)

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

Examples

```
require("dartR.data")
gl2gds(platypus.gl,snp_pos='ChromPos_Platypus_Chrom_NCBIv1',
snp_chr = 'Chrom_Platypus_Chrom_NCBIv1')
```

---

gl2genalex *Converts a genlight object into a format suitable for input to genalex*

Description

The output csv file contains the snp data and other relevant lines suitable for genalex. This script is a wrapper for genind2genalex (package poppr).

Usage

```
gl2genalex(
  x,
  outfile = "genalex.csv",
  outpath = tempdir(),
  overwrite = FALSE,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
outfile	File name of the output file (including extension) [default 'genalex.csv'].
outpath	Path where to save the output file [default tempdir()].
overwrite	If FALSE and filename exists, then the file will not be overwritten. Set this option to TRUE to overwrite the file [default FALSE].
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].

Author(s)

Custodian: Luis Mijangos, Author: Katrin Hohwieler, wrapper Arthur Georges (Post to https://groups.google.com/d/forum/dartr)

References

Peakall, R. and Smouse P.E. (2012) GenAlEx 6.5: genetic analysis in Excel. Population genetic software for teaching and research-an update. Bioinformatics 28, 2537-2539. http://bioinformatics.oxfordjournals.org/conten

Examples

```
gl2genalex(testset.gl, outfile='testset.csv')
```

---

gl2genepop *Converts a genlight object into genepop format (and file)*

Description

The genepop format is used by several external applications (for example Neestimator2 (http://www.molecularfisherieslaboratory.com.au/neestimator-software/). So the main idea is to create the genepop file and then run the other software externally. As a feature, the genepop file is also returned as an invisible data.frame by the function.

Usage

```
gl2genepop(
  x,
  outfile = "genepop.gen",
  outpath = tempdir(),
  pop_order = "alphabetic",
  output_format = "2_digits",
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
outfile	File name of the output file [default 'genepop.gen'].
outpath	Path where to save the output file. Use outpath=getwd() or outpath='.' when calling this function to direct output files to your working directory [default tempdir(), mandated by CRAN].
pop_order	Order of the output populations either "alphabetic" or a vector of population names in the order required by the user (see examples) [default "alphabetic"].
output_format	Whether to use a 2-digit format ("2_digits") or 3-digits format ("3_digits") [default "2_digits"].
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].

Value

Invisible data frame in genepop format

Author(s)

Custodian: Bernd Gruber (Post to https://groups.google.com/d/forum/dartr)

Examples

```
## Not run:
require("dartR.data")
# SNP data
geno <- gl2genepop(testset.gl[1:3,1:9])
head(geno)
test <- gl.filter.callrate(platypus.gl,threshold = 1)
popNames(test)
gl2genepop(test, pop_order = c("TENTERFIELD","SEVERN_ABOVE","SEVERN_BELOW"),
           output_format="3_digits")

## End(Not run)
```

---

gl2geno *Converts a genlight object to geno format from package LEA*

Description

The function converts a genlight object (SNP or presence/absence i.e. SilicoDArT data) into a file in the 'geno' and the 'lfmm' formats from (package LEA).

Usage

```
gl2geno(x, outfile = "gl_geno", outpath = tempdir(), verbose = NULL)
```

Arguments

x	Name of the genlight object containing the SNP or presence/absence (Silico-DArT) data [required].
outfile	File name of the output file [default 'gl_geno'].
outpath	Path where to save the output file [default tempdir(), mandated by CRAN]. Use outpath=getwd() or outpath='.' when calling this function to direct output files to your 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, unless specified using gl.set.verbosity].

Author(s)

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

Examples

```
# SNP data
gl2geno(testset.gl)
# Tag P/A data
gl2geno(testset.gs)
```

---

gl2gi *Converts a genlight object to genind object*

Description

Converts a genlight object to genind object

Usage

```
gl2gi(x, probar = FALSE, verbose = NULL)
```

Arguments

x	A genlight object [required].
probar	If TRUE, a progress bar will be displayed for long loops [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

This function uses a faster version of df2genind (from the adegenet package)

Value

A genind object, with all slots filled.

Author(s)

Bernd Gruber (Post to https://groups.google.com/d/forum/dartr)

---

gl2hiphop *Converts a genlight objects into hiphop format*

Description

This function exports genlight objects to the format used by the parentage assignment R package hiphop. Hiphop can be used for paternity and maternity assignment and outperforms conventional methods where closely related individuals occur in the pool of possible parents. The method compares the genotypes of offspring with any combination of potentials parents and scores the number of mismatches of these individuals at bi-allelic genetic markers (e.g. Single Nucleotide Polymorphisms).

Usage

```
gl2hiphop(x, verbose = NULL)
```

Arguments

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

Value

Dataframe containing all the genotyped individuals (offspring and potential parents) and their genotypes scored using bi-allelic markers.

Author(s)

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

References

Cockburn, A., Penalba, J.V.,Jaccoud, D.,Kilian, A., Brouwer, L.,Double, M.C., Margraf, N., Osmond, H.L., van de Pol, M. and Kruuk, L.E.B.(in revision). HIPHOP: improved paternity assignment among close relatives using a simple exclusion method for bi-allelic markers. Molecular Ecology Resources, DOI to be added upon acceptance

Examples

```
result <- gl2hiphop(testset.gl)
```

gl2phylip *Creates a Phylip input distance matrix from a genlight (SNP) {adegenet} object*

Description

This function calculates and returns a matrix of Euclidean distances between populations and produces an input file for the phylogenetic program Phylip (Joe Felsenstein).

Usage

```
gl2phylip(
  x,
  outfile = "phyinput.txt",
  outpath = tempdir(),
  bstrap = 1,
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP or presence/absence (SilicoDArT) data [required].
outfile	Name of the file to become the input file for phylip [default "phyinput.txt"].
outpath	Path where to save the output file [default tempdir(), mandated by CRAN]. Use outpath=getwd() or outpath='.' when calling this function to direct output files to your working directory.
bstrap	Number of bootstrap replicates [default 1].
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

Matrix of Euclidean distances between populations.

Author(s)

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

Examples

```
result <- gl2phylip(testset.gl, outfile='test.txt', bstrap=10)
```

gl2plink *Converts a genlight object into PLINK format*

Description

This function exports a genlight object into PLINK format and save it into a file. This function produces the following PLINK files: bed, bim, fam, ped and map.

Usage

```
gl2plink(
  x,
  plink_path = getwd(),
  bed_file = FALSE,
  outfile = "gl_plink",
  outpath = tempdir(),
  chr_format = "character",
  pos_cM = "0",
  ID_dad = "0",
  ID_mom = "0",
  sex_code = "unknown",
  phen_value = "0",
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
plink_path	Path of PLINK binary file [default getwd()].
bed_file	Whether create PLINK files .bed, .bim and .fam [default FALSE].
outfile	File name of the output file [default 'gl_plink'].
outpath	Path where to save the output file [default tempdir(), mandated by CRAN]. Use outpath=getwd() or outpath='.' when calling this function to direct output files to your working directory.
chr_format	Whether chromosome information is stored as 'numeric' or as 'character', see details [default 'character'].
pos_cM	A vector, with as many elements as there are loci, containing the SNP position in morgans or centimorgans [default '0'].
ID_dad	A vector, with as many elements as there are individuals, containing the ID of the father, '0' if father isn't in dataset [default '0'].
ID_mom	A vector, with as many elements as there are individuals, containing the ID of the mother, '0' if mother isn't in dataset [default '0'].

sex_code	A vector, with as many elements as there are individuals, containing the sex code ('male', 'female', 'unknown'). Sex information needs just to start with an "F" or "f" for females, with an "M" or "m" for males and with a "U", "u" or being empty if the sex is unknown [default 'unknown'].
phen_value	A vector, with as many elements as there are individuals, containing the phenotype value. '1' = control, '2' = case, '0' = unknown [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

To create PLINK files .bed, .bim and .fam (bed_file = TRUE), it is necessary to download the binary file of PLINK 1.9 and provide its path (plink_path). The binary file can be downloaded from: https://www.cog-genomics.org/plink/

After downloading, unzip the file, access the unzipped folder and move the binary file ("plink") to your working directory.

If you are using a Mac, you might need to open the binary first to grant access to the binary.

The chromosome of each SNP can be a character or numeric. The chromosome information for unmapped SNPS is coded as 0. Family ID is taken from x$pop. Within-family ID (cannot be '0') is taken from indNames(x). Variant identifier is taken from locNames(x). SNP position is taken from the accessor x$position. Chromosome name is taken from the accessor x$chromosome Note that if names of populations or individuals contain spaces, they are replaced by an underscore "_".

If you like to use chromosome information when converting to plink format and your chromosome names are not from human, you need to change the chromosome names as 'contig1', 'contig2', etc. as described in the section "Nonstandard chromosome IDs" in the following link: https://www.cog-genomics.org/plink/1.9/input

Author(s)

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

References

Purcell, Shaun, et al. 'PLINK: a tool set for whole-genome association and population-based linkage analyses.' The American journal of human genetics 81.3 (2007): 559-575.

Examples

```
require("dartR.data")
test <- platypus.gl
# assigning SNP position
test$position <- test$other$loc.metrics$ChromPos_Platypus_Chrom_NCBIv1
# assigning a dummy name for chromosomes
test$chromosome <- as.factor("1")
gl2plink(test)
```

`gl2related` *Converts a genlight object to format suitable to be run with Coancestry*

Description

The output txt file contains the SNP data and an additional column with the names of the individual. The file then can be used and loaded into coancestry or - if installed - run with the related package. Be aware the related package was crashing in previous versions, but in general is using the same code as coancestry and therefore should have identical results. Also running coancestry with thousands of SNPs via the GUI seems to be not reliable and therefore for comparisons between coancestry and related we suggest to use the command line version of coancestry.

Usage

```
gl2related(
  x,
  outfile = "related.txt",
  outpath = tempdir(),
  save = TRUE,
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object containing the SNP data [required].
`outfile`	File name of the output file (including extension) [default 'related.txt'].
`outpath`	Path where to save the output file [default tempdir()].
`save`	A switch if you want to save the file or not. This might be useful for someone who wants to use the coancestry function to calculate relatedness and not export to coancestry. See the example below [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, unless specified using gl.set.verbosity].

Value

A data.frame that can be used to run with the related package

Author(s)

Bernd Gruber (bugs? Post to https://groups.google.com/d/forum/dartr)

References

Jack Pew, Jinliang Wang, Paul Muir and Tim Frasier (2014). related: related: an R package for analyzing pairwise relatedness data based on codominant molecular markers. R package version 0.8/r2. https://R-Forge.R-project.org/projects/related/

Examples

```
gtd <- gl2related(bandicoot.gl[1:10,1:20], save=FALSE)
## Not run:
##running with the related package
#install.packages('related', repos='http://R-Forge.R-project.org')
library(related)
coan <- coancestry(gtd, wang=1)
head(coan$relatedness)
##check ?coancestry for information how to use the function.

## End(Not run)
```

---

gl2sa *Converts genlight objects to the format used in the SNPassoc package*

Description

This function exports a genlight object into a SNPassoc object. See package SNPassoc for details. This function needs package SNPassoc. At the time of writing (August 2020) the package was no longer available from CRAN. To install the package check their github repository. https://github.com/isglobal-brge/SNPassoc and/or use `install_github('isglobal-brge/SNPassoc')` to install the function and uncomment the function code.

Usage

```
gl2sa(x, verbose = NULL, installed = FALSE)
```

Arguments

x	Name of the genlight object containing the SNP data [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].
installed	Switch to run the function once SNPassoc package i s installed [default FALSE].

Value

Returns an object of class 'snp' to be used with SNPassoc.

Author(s)

Bernd Guber (Post to https://groups.google.com/d/forum/dartr)

References

Gonzalez, J.R., Armengol, L., Sol?, X., Guin?, E., Mercader, J.M., Estivill, X. and Moreno, V. (2017). SNPassoc: an R package to perform whole genome association studies. Bioinformatics 23:654-655.

gl2sfs *Converts a genlight object into a sfs input file*

Description

The output of this function is suitable for analysis in fastsimcoal2 or dada.

Usage

```
gl2sfs(
  x,
  n.invariant.tags = 0,
  outfile_root = "gl2sfs",
  outpath = tempdir(),
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
n.invariant.tags	Number of invariant sites[default 0].
outfile_root	The root of the name of the output file [default "gl2sfs"].
outpath	Path where to save the output file [default 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

It saves a derived sfs, assuming that the reference allele is the ancestral, and a MAF sfs.

At this stage this function caters only for diploid organisms, for samples from one population only, and for genotypes without missing data. Note that sfs uses frequencies considered **independent**, data are assumed to be from independent (i.e. not linked) loci. This means that only one site per tag should be considered 9i.e. secondaries should be removed). If no monomorphic site estimates is provided (with `n.invariant.tags`), the sfs will only include the number of monomorphic sites in the data (but this will be a biased estimates as it doesn't take into account the invariant tags that have not been included. This will affect parameter estimates in the analyses). Note that the number of invariant tags can be estimated with `gl.report.secondaries`. In a limited number of cases, ascertainment bias can be explicitly modelled in fastsimcoal2. See fastsimcoal2 manual for details.

It expects a dartR formatted genlight object, but it should also work with other genlight objects.

Value

Deprecated. Please use gl.sfs instead.

Author(s)

Custodian: Carlo Pacioni (Post to https://groups.google.com/d/forum/dartr)

References

Excoffier L., Dupanloup I., Huerta-Sánchez E., Sousa V. C. and Foll M. (2013) Robust demographic inference from genomic and SNP data. PLoS genetics 9(10)

See Also

gl.report.heterozygosity, gl.report.secondaries, utils.n.var.invariant

---

gl2shp *Converts a genlight object to ESRI shapefiles or kml files*

Description

This function exports coordinates in a genlight object to a point shape file (including also individual meta data if available). Coordinates are provided under x@other$latlon and assumed to be in WGS84 coordinates, if not proj4 string is provided.

Usage

```
gl2shp(
  x,
  type = "shp",
  proj4 = "+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs",
  outfile = "gl",
  outpath = tempdir(),
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data and location data, lat longs [required].
type	Type of output 'kml' or 'shp' [default 'shp'].
proj4	Proj4string of data set (see spatialreference.org for projections) [default WGS84].
outfile	Name (path) of the output shape file [default 'gl']. shp extension is added automatically.
outpath	Path where to save the output file [default tempdir(), mandated by CRAN]. Use outpath=getwd() or outpath='.' when calling this function to direct output files to your 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].

Value

returns a SpatVector file

Author(s)

Bernd Guber (Post to https://groups.google.com/d/forum/dartr)

Examples

```
out <- gl2shp(testset.gl)
```

---

gl2snapp — *Converts a genlight object to nexus format suitable for phylogenetic analysis by SNAPP (via BEAUti)*

Description

The output nexus file contains the SNP data and relevant PAUP command lines suitable for BEAUti.

Usage

```
gl2snapp(x, outfile = "snapp.nex", outpath = tempdir(), verbose = NULL)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
outfile	File name of the output file (including extension) [default "snapp.nex"].
outpath	Path where to save the output file [default tempdir(), mandated by CRAN]. Use outpath=getwd() or outpath='.' when calling this function to direct output files to your 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].

Author(s)

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

References

Bryant, D., Bouckaert, R., Felsenstein, J., Rosenberg, N.A. and RoyChoudhury, A. (2012). Inferring species trees directly from biallelic genetic markers: bypassing gene trees in a full coalescent analysis. Molecular Biology and Evolution 29:1917-1932.

Examples

```
gl2snapp(testset.gl)
```

`gl2structure` *Converts a genlight object to STRUCTURE formated files*

Description

This function exports genlight objects to STRUCTURE formatted files (be aware there is a gl2faststructure version as well). It is based on the code provided by Lindsay Clark (see https://github.com/lvclark/R_genetics_conv) and this function is basically a wrapper around her numeric2structure function. See also: Lindsay Clark. (2017, August 22). lvclark/R_genetics_conv: R_genetics_conv 1.1 (Version v1.1). Zenodo: doi.org/10.5281/zenodo.846816.

Usage

```
gl2structure(
  x,
  indNames = NULL,
  addcolumns = NULL,
  ploidy = 2,
  exportMarkerNames = TRUE,
  outfile = "gl.str",
  outpath = tempdir(),
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object containing the SNP data and location data, lat longs [required].
`indNames`	Specify individuals names to be added [if NULL, defaults to indNames(x)].
`addcolumns`	Additional columns to be added before genotypes [default NULL].
`ploidy`	Set the ploidy [defaults 2].
`exportMarkerNames`	If TRUE, locus names locNames(x) will be included [default TRUE].
`outfile`	File name of the output file (including extension) [default "gl.str"].
`outpath`	Path where to save the output file [default tempdir(), mandated by CRAN]. Use outpath=getwd() or outpath='.' when calling this function to direct output files to your 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].

Author(s)

Bernd Gruber (wrapper) and Lindsay V. Clark [lvclark@illinois.edu]

Examples

```
#not run here
#gl2structure(testset.gl)
```

`gl2svdquartets` *Converts a genlight object to nexus format PAUP SVDquartets*

Description

The output nexus file contains the SNP data in one of two forms, depending upon what you regard as most appropriate. One form, that used by Chifman and Kubatko, has two lines per individual, one providing the reference SNP the second providing the alternate SNP (method=1).

Usage

```
gl2svdquartets(
  x,
  outfile = "svd.nex",
  outpath = tempdir(),
  method = 2,
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object containing the SNP data or tag P/A data [required].
`outfile`	File name of the output file (including extension) [default 'svd.nex'].
`outpath`	Path where to save the output file [default tempdir(), mandated by CRAN]. Use outpath=getwd() when calling this function or set.tempdir <- getwd() elsewhere in your script to direct output files to your working directory.
`method`	Method = 1, nexus file with two lines per individual; method = 2, nexus file with one line per individual, ambiguity codes for SNP genotypes, 0 or 1 for presence/absence data [default 2].
`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

A second form, recommended by Dave Swofford, has a single line per individual, resolving heterozygous SNPs by replacing them with standard ambiguity codes (method=2).

If the data are tag presence/absence, then method=2 is assumed.

Author(s)

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

References

Chifman, J. and L. Kubatko. 2014. Quartet inference from SNP data under the coalescent. Bioinformatics 30: 3317-3324

Examples

```
gg <- testset.gl[1:20,1:100]
gg@other$loc.metrics <- gg@other$loc.metrics[1:100,]
gl2svdquartets(gg)
```

`gl2treemix` *Converts a genlight object to a treemix input file*

Description

The output file contains the SNP data in the format expected by treemix – see the treemix manual. The file will be gzipped before in order to be recognised by treemix. Plotting functions provided with treemix will need to be sourced from the treemix download page.

Usage

```
gl2treemix(
  x,
  outfile = "treemix_input.gz",
  outpath = tempdir(),
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight object containing the SNP data [required].
`outfile`	File name of the output file (including gz extension) [default 'treemix_input.gz'].
`outpath`	Path where to save the output file [default tempdir(), mandated by CRAN]. Use outpath=getwd() when calling this function or set.tempdir <- getwd() elsewhere in your script to direct output files to your 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].

Author(s)

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

References

Pickrell and Pritchard (2012). Inference of population splits and mixtures from genome-wide allele frequency data. PLoS Genetics https://doi.org/10.1371/journal.pgen.1002967

Examples

```
gl2treemix(testset.gl, outpath=tempdir())
```

---

gl2vcf *Converts a genlight object into vcf format*

Description

This function exports a genlight object into VCF format and save it into a file.

Usage

```
gl2vcf(
  x,
  plink_path = getwd(),
  outfile = "gl_vcf",
  outpath = tempdir(),
  snp_pos = "0",
  snp_chr = "0",
  chr_format = "character",
  pos_cM = "0",
  ID_dad = "0",
  ID_mom = "0",
  sex_code = "unknown",
  phen_value = "0",
  verbose = NULL
)
```

Arguments

x	Name of the genlight object containing the SNP data [required].
plink_path	Path of PLINK binary file [default getwd())].
outfile	File name of the output file [default 'gl_vcf'].
outpath	Path where to save the output file [default tempdir(), mandated by CRAN]. Use outpath=getwd() or outpath='.' when calling this function to direct output files to your working directory.
snp_pos	Field name from the slot loc.metrics where the SNP position is stored [default '0'].
snp_chr	Field name from the slot loc.metrics where the chromosome of each is stored [default '0'].
chr_format	Whether chromosome information is stored as 'numeric' or as 'character', see details [default 'character'].
pos_cM	A vector, with as many elements as there are loci, containing the SNP position in morgans or centimorgans [default '0'].

ID_dad	A vector, with as many elements as there are individuals, containing the ID of the father, '0' if father isn't in dataset [default '0'].
ID_mom	A vector, with as many elements as there are individuals, containing the ID of the mother, '0' if mother isn't in dataset [default '0'].
sex_code	A vector, with as many elements as there are individuals, containing the sex code ('male', 'female', 'unknown') [default 'unknown'].
phen_value	A vector, with as many elements as there are individuals, containing the phenotype value. '1' = control, '2' = case, '0' = unknown [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 requires to download the binary file of PLINK 1.9 and provide its path (plink_path). The binary file can be downloaded from: https://www.cog-genomics.org/plink/

The chromosome information for unmapped SNPS is coded as 0. Family ID is taken from x$pop Within-family ID (cannot be '0') is taken from indNames(x) Variant identifier is taken from locNames(x)

Author(s)

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

References

Danecek, P., Auton, A., Abecasis, G., Albers, C. A., Banks, E., DePristo, M. A., ... & 1000 Genomes Project Analysis Group. (2011). The variant call format and VCFtools. Bioinformatics, 27(15), 2156-2158.

Examples

```
## Not run:
require("dartR.data")
gl2vcf(platypus.gl,snp_pos='ChromPos_Platypus_Chrom_NCBIv1',
 snp_chr = 'Chrom_Platypus_Chrom_NCBIv1')

## End(Not run)
```

---

interactive_reference *Shiny app for the input of the reference table for the simulations*

Description

Shiny app for the input of the reference table for the simulations

Usage

```
interactive_reference()
```

Author(s)

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

---

`interactive_sim_run` *Shiny app for the input of the simulations variables*

Description

Shiny app for the input of the simulations variables

Usage

```
interactive_sim_run()
```

Author(s)

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

---

`is.fixed` *Tests if two populations are fixed at a given locus*

Description

This script compares two percent allele frequencies and reports TRUE if they represent a fixed difference, FALSE otherwise.

Usage

```
is.fixed(s1, s2, tloc = 0)
```

Arguments

`s1`	Percentage SNP allele or sequence tag frequency for the first population [required].
`s2`	Percentage SNP allele or sequence tag frequency for the second population [required].
`tloc`	Threshold value for tolerance in when a difference is regarded as fixed [default 0].

Details

A fixed difference at a locus occurs when two populations share no alleles, noting that SNPs are biallelic (ploidy=2). Tolerance in the definition of a fixed difference is provided by the t parameter. For example, t=0.05 means that SNP allele frequencies of 95,5 and 5,95 percent will be reported as fixed (TRUE).

Value

TRUE (fixed difference) or FALSE (alleles shared) or NA (one or both s1 or s2 missing)

Author(s)

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

See Also

gl.fixed.diff

Examples

```
is.fixed(s1=100, s2=0, tloc=0)
is.fixed(96, 4, tloc=0.05)
```

---

platy *Example data set as text file to be imported into a genlight object*

Description

Check ?read.genetable in pacakge PopGenReport for details on the format.

Format

csv

Author(s)

Bernd Gruber (bugs? Post to https://groups.google.com/d/forum/dartr

Examples

```
library(PopGenReport)
read.csv( paste(.libPaths()[1],'/dartR/extdata/platy.csv',sep='' ))
platy <- read.genetable( paste(.libPaths()[1],'/dartR/extdata/platy.csv',
sep='' ), ind=1, pop=2, lat=3, long=4, other.min=5, other.max=6,
oneColPerAll=FALSE, sep='/')
platy.gl <- gi2gl(platy, parallel=FALSE)
df.loc <- data.frame(RepAvg = runif(nLoc(platy.gl)), CallRate = 1)
platy.gl@other$loc.metrics <- df.loc
gl.report.reproducibility(platy.gl)
```

possums.gl *A simulated genlight object created to run a landscape genetic example*

Description

This a test data set to run a landscape genetics example. It contains 10 populations of 30 individuals each and each individual has 300 loci. There are no covariates for individuals or loci.

Usage

```
possums.gl
```

Format

genlight object

Author(s)

Bernd Gruber (bugs? Post to https://groups.google.com/d/forum/dartr

rbind.dartR *adjust rbind for dartR*

Description

rbind is a bit lazy and does not take care for the metadata (so data in the other slot is lost). You can get most of the loci metadata back using gl.compliance.check.

Usage

```
## S3 method for class 'dartR'
rbind(...)
```

Arguments

...	list of dartR objects

testset.gl — *A genlight object created via the gl.read.dart function*

Description

This is a test data set on turtles. 250 individuals, 255 loci in >30 populations.

Usage

```
testset.gl
```

Format

genlight object

Author(s)

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

testset.gs — *A genlight object created via the gl.read.silicodart function*

Description

This is a test data set on turtles. 218 individuals, 255 loci in >30 populations.

Usage

```
testset.gs
```

Format

genlight object

Author(s)

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

testset_metadata — *Metadata file. Can be integrated via the dart2genlight function.*

Description

Metadata file. Can be integrated via the dart2genlight function.

Format

csv

Author(s)

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

testset_pop_recode — *Recode file to be used with the function.*

Description

This test data set is provided to show a typical recode file format.

Format

csv

Author(s)

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

testset_SNPs_2Row — *Testfile in DArT format (as provided by DArT)*

Description

This test data set is provided to show a typical DArT file format. Can be used to create a genlight object using the read.dart function.

Format

csv

Author(s)

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

theme_dartR *dartR theme*

Description

This is the theme used as default for dartR plots. This function controls all non-data display elements in the plots.

Usage

```
theme_dartR(
  base_size = 11,
  base_family = "",
  base_line_size = base_size/22,
  base_rect_size = base_size/22
)
```

Arguments

base_size	base font size, given in pts.
base_family	base font family
base_line_size	base size for line elements
base_rect_size	base size for rect elements

Examples

```
#ggplot(data.frame(dummy=rnorm(1000)),aes(dummy)) +
#geom_histogram(binwidth=0.1) + theme_dartR()
```

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 = 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")
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 = 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_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")
```

utils.basic.stats *Calculates mean observed heterozygosity, mean expected heterozygosity and Fis per locus, per population and Fst across all populations*

Description

This is a re-implementation of `hierfstat::basics.stats` specifically for genlight objects. Formula (and hence results) match exactly the original version of `hierfstat::basics.stats` (although Dstp and Fstp are not currently computed) but it is much faster.

Usage

```
utils.basic.stats(x)
```

Arguments

x	A genlight object containing the SNP genotypes [required].

Value

A list with with the statistics for each population

Author(s)

Luis Mijangos and Carlo Pacioni (bugs? Post to https://groups.google.com/d/forum/dartr)

Examples

```
require("dartR.data")
out <- utils.basic.stats(platypus.gl)
```

utils.check.datatype *Utility function to check the class of an object passed to a function*

Description

Most functions require access to a genlight object, dist matrix, data matrix or fixed difference list (fd), and this function checks that a genlight object or one of the above has been passed, whether the genlight object is a SNP dataset or a SilicoDArT object, and reports back if verbosity is >=2.

Usage

```
utils.check.datatype(
  x,
  accept = c("genlight", "SNP", "SilicoDArT"),
  verbose = NULL
)
```

Arguments

x	Name of the genlight object, dist matrix, data matrix, glPCA, or fixed difference list (fd) [required].
accept	Vector containing the classes of objects that are to be accepted [default c('genlight','SNP','SilicoDArT'].
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

This function checks the class of passed object and sets the datatype to 'SNP', 'SilicoDArT', 'dist', 'mat', or class[1](x) as appropriate.

Note also that this function checks to see if there are individuals or loci scored as all missing (NA) and if so, issues the user with a warning.

Note: One and only one of gl.check, fd.check, dist.check or mat.check can be TRUE.

Value

datatype, 'SNP' for SNP data, 'SilicoDArT' for P/A data, 'dist' for a distance matrix, 'mat' for a data matrix, 'glPCA' for an ordination file, or class(x)[1].

Author(s)

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

Examples

```
datatype <- utils.check.datatype(testset.gl)
datatype <- utils.check.datatype(as.matrix(testset.gl),accept='matrix')
fd <- gl.fixed.diff(testset.gl)
datatype <- utils.check.datatype(fd,accept='fd')

datatype <- utils.check.datatype(testset.gl)
```

---

utils.dart2genlight *Converts DarT to genlight*

Description

Converts a DArT file (read via read.dart) into an genlight object adegenet. Internal function called by gl.read.dart

Usage

```
utils.dart2genlight(
  dart,
  ind.metafile = NULL,
  covfilename = NULL,
  probar = TRUE,
  verbose = NULL
)
```

Arguments

dart	A dart object created via read.dart [required].
ind.metafile	Optional file in csv format with metadata for each individual (see details for explanation) [default NULL].
covfilename	Depreciated, use parameter ind.metafile.
probar	Show progress bar [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 NULL].

Details

The ind.metadata file needs to have very specific headings. First a heading called id. Here the ids have to match the ids in the dart object `colnames(dart[[4]])`. The following column headings are optional. pop: specifies the population membership of each individual. lat and lon specify spatial coordinates (in decimal degrees WGS1984 format). Additional columns with individual metadata can be imported (e.g. age, gender).

Value

A genlight object. Including all available slots are filled. loc.names, ind.names, pop, lat, lon (if provided via the ind.metadata file)

---

utils.dist.binary *Calculates a distance matrix for individuals defined in a dartR genlight object using binary P/A data (SilicoDArT)*

Description

This script calculates various distances between individuals based on sequence tag Presence/Absence data.

Usage

```
utils.dist.binary(
  x,
  method = "simple",
  scale = FALSE,
  swap = FALSE,
  output = "dist",
  verbose = NULL
)
```

Arguments

x	Name of the genlight containing the genotypes [required].
method	Specify distance measure [default simple].
scale	If TRUE and method='euclidean', the distance will be scaled to fall in the range [0,1] [default FALSE].
swap	If TRUE and working with presence-absence data, then presence (no disrupting mutation) is scored as 0 and absence (presence of a disrupting mutation) is scored as 1 [default FALSE].
output	Specify the format and class of the object to be returned, dist for a object of class dist, matrix for an object of class matrix [default "dist"].
verbose	Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2].

Details

The distance measure can be one of:

- Euclidean – Euclidean Distance applied to cartesian coordinates defined by the loci, scored as 0 or 1. Presence and absence equally weighted.
- simple – simple matching, both 1 or both 0 = 0; one 1 and the other 0 = 1. Presence and absence equally weighted.
- Jaccard – ignores matching 0, both 1 = 0; one 1 and the other 0 = 1. Absences could be for different reasons.
- Bray-Curtis – both 0 = 0; both 1 = 2; one 1 and the other 0 = 1. Absences could be for different reasons. Sometimes called the Dice or Sorensen distance.

One might choose to disregard or downweight absences in comparison with presences because the homology of absences is less clear (mutation at one or the other, or both restriction sites). Your call.

Value

An object of class 'dist' or 'matrix' giving distances between individuals

Author(s)

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

Examples

```
D <- utils.dist.binary(testset.gs, method='Jaccard')
D <- utils.dist.binary(testset.gs, method='Euclidean',scale=TRUE)

D <- utils.dist.binary(testset.gs, method='Simple')
```

`utils.dist.ind.snp` *Calculates a distance matrix for individuals defined in a dartR genlight object using SNP data (DArTseq)*

Description

This script calculates various distances between individuals based on SNP genotypes.

Usage

```
utils.dist.ind.snp(
  x,
  method = "Euclidean",
  scale = FALSE,
  output = "dist",
  verbose = NULL
)
```

Arguments

`x`	Name of the genlight containing the genotypes [required].
`method`	Specify distance measure [default Euclidean].
`scale`	If TRUE and method='Euclidean', the distance will be scaled to fall in the range [0,1] [default FALSE].
`output`	Specify the format and class of the object to be returned, dist for a object of class dist, matrix for an object of class matrix [default "dist"].
`verbose`	Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2].

Details

The distance measure can be one of:

- Euclidean – Euclidean Distance applied to Cartesian coordinates defined by the loci, scored as 0, 1 or 2.
- Simple – simple mismatch, 0 where no alleles are shared, 1 where one allele is shared, 2 where both alleles are shared.

- Absolute – absolute mismatch, 0 where no alleles are shared, 1 where one or both alleles are shared.
- Czekanowski (or Manhattan) calculates the city block metric distance by summing the scores on each axis (locus).

Value

An object of class 'dist' or 'matrix' giving distances between individuals

Author(s)

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

Examples

```
D <- utils.dist.ind.snp(testset.gl, method='Manhattan')
D <- utils.dist.ind.snp(testset.gl, method='Euclidean',scale=TRUE)

D <- utils.dist.ind.snp(testset.gl, method='Simple')
```

---

utils.flag.start *A utility script to flag the start of a script*

Description

A utility script to flag the start of a script

Usage

```
utils.flag.start(func = NULL, build = NULL, verbosity = NULL)
```

Arguments

func	Name of the function that is starting [required].
build	Name of the build [default NULL].
verbosity	Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2].

Value

calling function name

Author(s)

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

`utils.hamming` *Calculates the Hamming distance between two DArT trimmed DNA sequences*

Description

Hamming distance is calculated as the number of base differences between two sequences which can be expressed as a count or a proportion. Typically, it is calculated between two sequences of equal length. In the context of DArT trimmed sequences, which differ in length but which are anchored to the left by the restriction enzyme recognition sequence, it is sensible to compare the two trimmed sequences starting from immediately after the common recognition sequence and terminating at the last base of the shorter sequence.

Usage

```
utils.hamming(str1, str2, r = 4)
```

Arguments

`str1`	String containing the first sequence [required].
`str2`	String containing the second sequence [required].
`r`	Number of bases in the restriction enzyme recognition sequence [default 4].

Details

The Hamming distance between the rows of a matrix can be computed quickly by exploiting the fact that the dot product of two binary vectors x and (1-y) counts the corresponding elements that are different between x and y. This matrix multiplication can also be used for matrices with more than two possible values, and different types of elements, such as DNA sequences.

The function calculates the Hamming distance between all columns of a matrix X, or two matrices X and Y. Again matrix multiplication is used, this time for counting, between two columns x and y, the number of cases in which corresponding elements have the same value (e.g. A, C, G or T). This counting is done for each of the possible values individually, while iteratively adding the results. The end result of the iterative adding is the sum of all corresponding elements that are the same, i.e. the inverse of the Hamming distance. Therefore, the last step is to subtract this end result H from the maximum possible distance, which is the number of rows of matrix X.

If the two DNA sequences are of differing length, the longer is truncated. The initial common restriction enzyme recognition sequence is ignored.

The algorithm is that of Johann de Jong https://johanndejong.wordpress.com/2015/10/02/faster-hamming-distance-in-r-2/

Value

Hamming distance between the two strings

Author(s)

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

`utils.het.pop` *Calculates expected mean expected heterozygosity per population*

Description

Calculates expected mean expected heterozygosity per population

Usage

```
utils.het.pop(x)
```

Arguments

x	A genlight object containing the SNP genotypes [required].

Value

A vector with the mean expected heterozygosity for each population

Author(s)

Bernd Gruber & Luis Mijangos (bugs? Post to https://groups.google.com/d/forum/dartr)

Examples

```
out <- utils.het.pop(testset.gl)
```

`utils.jackknife` *Conducts jackknife resampling using a genlight object*

Description

Jackknife resampling is a statistical procedure where for a dataset of sample size n, subsamples of size n-1 are used to compute a statistic. The collection of the values obtained can be used to evaluate the variability around the point estimate. This function can take the loci, the individuals or the populations as units over which to conduct resampling.

boldNote that when n is very small, jackknife resampling is not recommended.

Parallel computation is implemented. The argument coden.cores indicates the number of core to use. If "auto" [default], it will use all but one available cores. If the number of units is small (e.g. a few populations), there is not real advantage in using parallel computation. On the other hand, if the number of units is large (e.g. thousands of loci), even with parallel computation, this function can be very slow.

Usage

```
utils.jackknife(
  x,
  FUN,
  unit = "loc",
  recalc = FALSE,
  mono.rm = FALSE,
  n.cores = "auto",
  verbose = NULL,
  ...
)
```

Arguments

x	Name of the genlight object containing SNP genotypes [required].
FUN	the name of the function to be used to calculate the statistic
unit	The unit to use for resampling. One of c("loc", "ind", "pop"): loci, individuals or populations
recalc	Recalculate the locus metadata statistics [default FALSE].
mono.rm	Remove monomorphic loci [default FALSE].
n.cores	The number of cores to use. If "auto" [default], it will use all but one available cores.
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].
...	any additional arguments to be passed to FUN

Value

A list of length n where each element is the output of FUN

Author(s)

Custodian: Carlo Pacioni – Post to https://groups.google.com/d/forum/dartr

Examples

```
require("dartR.data")
platMod.gl <- gl.filter.allna(platypus.gl)
chk.pop <- utils.jackknife(x=platMod.gl, FUN="gl.alf", unit="pop",
recalc = FALSE, mono.rm = FALSE, n.cores = 1, verbose=0)
```

`utils.n.var.invariant` *A utility script to calculate the number of variant and invariant sites by locus*

Description

Calculate the number of variant and invariant sites by locus and add them as columns in `loc.metrics`. This can be useful to conduct further filtering, for example where only loci with secondaries are wanted for phylogenetic analyses.

Usage

```
utils.n.var.invariant(x, verbose = NULL)
```

Arguments

`x`	Name of the genlight object containing the SNP data [required].
`verbose`	Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default NULL].

Details

Invariant sites are the sites (nucleotide) that are not polymorphic. When the locus metadata supplied by DArT includes the sequence of the allele (`TrimmedSequence`), it is used by this function to estimate the number of sites that were sequenced in each tag (read). This script then subtracts the number of polymorphic sites. The length of the trimmed sequence (lenTrimSeq), the number of variant (n.variant) and invariant (n.invariant) sites are the added to the table in `gl@others$loc.metrics`.

NOTE: It is important to realise that this function correctly estimates the number of variant and invariant sites only when it is executed on `genlight` objects before secondaries are removed.

Value

The modified genlight object.

Author(s)

Carlo Pacioni (Post to https://groups.google.com/d/forum/dartr)

See Also

`gl.filter.secondaries`,`gl.report.heterozygosity`

Examples

```
require("dartR.data")
out <- utils.n.var.invariant(platypus.gl)
```

`utils.outflank` *OutFLANK: An Fst outlier approach by Mike Whitlock and Katie Lotterhos, University of British Columbia.*

Description

This function is the original implementation of Outflank by Whitlock and Lotterhos. dartR simply provides a convenient wrapper around their functions and an easier install being an r package (for information please refer to their github repository)

Usage

```
utils.outflank(
  FstDataFrame,
  LeftTrimFraction = 0.05,
  RightTrimFraction = 0.05,
  Hmin = 0.1,
  NumberOfSamples,
  qthreshold = 0.05
)
```

Arguments

`FstDataFrame` A data frame that includes a row for each locus, with columns as follows:

- $LocusName: a character string that uniquely names each locus.
- $FST: Fst calculated for this locus. (Kept here to report the unbased Fst of the results)
- $T1: The numerator of the estimator for Fst (necessary, with $T2, to calculate mean Fst)
- $T2: The denominator of the estimator of Fst
- $FSTNoCorr: Fst calculated for this locus without sample size correction. (Used to find outliers)
- $T1NoCorr: The numerator of the estimator for Fst without sample size correction (necessary, with $T2, to calculate mean Fst)
- $T2NoCorr: The denominator of the estimator of Fst without sample size correction
- $He: The heterozygosity of the locus (used to screen out low heterozygosity loci that have a different distribution)

`LeftTrimFraction`
The proportion of loci that are trimmed from the lower end of the range of Fst before the likelihood funciton is applied [default 0.05].

`RightTrimFraction`
The proportion of loci that are trimmed from the upper end of the range of Fst before the likelihood funciton is applied [default 0.05].

`Hmin` The minimum heterozygosity required before including calculations from a locus [default 0.1].

`NumberOfSamples` The number of spatial locations included in the data set.

`qthreshold` The desired false discovery rate threshold for calculating q-values [default 0.05].

Details

This method looks for Fst outliers from a list of Fst's for different loci. It assumes that each locus has been genotyped in all populations with approximately equal coverage.

OutFLANK estimates the distribution of Fst based on a trimmed sample of Fst's. It assumes that the majority of loci in the center of the distribution are neutral and infers the shape of the distribution of neutral Fst using a trimmed set of loci. Loci with the highest and lowest Fst's are trimmed from the data set before this inference, and the distribution of Fst df/(mean Fst) is assumed to'follow a chi-square distribution. Based on this inferred distribution, each locus is given a q-value based on its quantile in the inferred null'distribution.

The main procedure is called OutFLANK – see comments in that function immediately below for input and output formats. The other functions here are necessary and must be uploaded, but are not necessarily needed by the user directly.

Steps:

Value

The function returns a list with seven elements:

- FSTbar: the mean FST inferred from loci not marked as outliers
- FSTNoCorrbar: the mean FST (not corrected for sample size -gives an upwardly biased estimate of FST)
- dfInferred: the inferred number of degrees of freedom for the chi-square distribution of neutral FST
- numberLowFstOutliers: Number of loci flagged as having a significantly low FST (not reliable)
- numberHighFstOutliers: Number of loci identified as having significantly high FST
- results: a data frame with a row for each locus. This data frame includes all the original columns in the data set, and six new ones:
  - $indexOrder (the original order of the input data set),
  - $GoodH (Boolean variable which is TRUE if the expected heterozygosity is greater than the Hemin set by input),
  - $OutlierFlag (TRUE if the method identifies the locus as an outlier, FALSE otherwise), and
  - $q (the q-value for the test of neutrality for the locus)
  - $pvalues (the p-value for the test of neutrality for the locus)
  - $pvaluesRightTail the one-sided (right tail) p-value for a locus

Author(s)

Bernd Gruber (bugs? Post to https://groups.google.com/d/forum/dartr); original implementation of Whitlock & Lotterhos

---

`utils.outflank.MakeDiploidFSTMat`
Creates OutFLANK input file from individual genotype info.

Description

Creates OutFLANK input file from individual genotype info.

Usage

```
utils.outflank.MakeDiploidFSTMat(SNPmat, locusNames, popNames)
```

Arguments

`SNPmat`	This is an array of genotypes with a row for each individual. There should be a column for each SNP, with the number of copies of the focal allele (0, 1, or 2) for that individual. If that individual is missing data for that SNP, there should be a 9, instead.
`locusNames`	A list of names for each SNP locus. There should be the same number of locus names as there are columns in SNPmat.
`popNames`	A list of population names to give location for each individual. Typically multiple individuals will have the same popName. The list popNames should have the same length as the number of rows in SNPmat.

Value

Returns a data frame in the form needed for the main OutFLANK function.

`utils.outflank.plotter`
Plotting functions for Fst distributions after OutFLANK

Description

This function takes the output of OutFLANK as input with the OFoutput parameter. It plots a histogram of the FST (by default, the uncorrected FSTs used by OutFLANK) of loci and overlays the inferred null histogram.

Usage

```
utils.outflank.plotter(
  OFoutput,
  withOutliers = TRUE,
  NoCorr = TRUE,
  Hmin = 0.1,
  binwidth = 0.005,
  Zoom = FALSE,
  RightZoomFraction = 0.05,
  titletext = NULL
)
```

Arguments

OFoutput	The output of the function OutFLANK()
withOutliers	Determines whether the loci marked as outliers (with $OutlierFlag) are included in the histogram.
NoCorr	Plots the distribution of FSTNoCorr when TRUE. Recommended, because this is the data used by OutFLANK to infer the distribution.
Hmin	The minimum heterozygosity required before including a locus in the plot.
binwidth	The width of bins in the histogram.
Zoom	If Zoom is set to TRUE, then the graph will zoom in on the right tail of the distirbution (based on argument RightZoomFraction)
RightZoomFraction	Used when Zoom = TRUE. Defines the proportion of the distribution to plot.
titletext	Allows a test string to be printed as a title on the graph

Value

produces a histogram of the FST

---

utils.read.dart *Imports DarT data to R*

Description

Internal function called by gl.read.dart

Usage

```
utils.read.dart(
  filename,
  nas = "-",
  topskip = NULL,
  lastmetric = "RepAvg",
```

```
  service_row = 1,
  plate_row = 3,
  verbose = NULL
)
```

Arguments

filename	Path to file (csv file only currently) [required].
nas	A character specifying NAs [default '-'].
topskip	A number specifying the number of rows to be skipped. If not provided the number of rows to be skipped are 'guessed' by the number of rows with '*' at the beginning [default NULL].
lastmetric	Specifies the last non genetic column [default 'RepAvg']. Be sure to check if that is true, otherwise the number of individuals will not match. You can also specify the last column by a number.
service_row	The row number in which the information of the DArT service is contained [default 1].
plate_row	The row number in which the information of the plate location is contained [default 3].
verbose	Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log ; 3, progress and results summary; 5, full report [default NULL].

Value

A list of length 5. #dart format (one or two rows) #individuals, #snps, #non genetic metrics, #genetic data (still two line format, rows=snps, columns=individuals)

---

utils.recalc.avgpic *A utility script to recalculate the OneRatioRef, OneRatioSnp, PICRef, PICSnp, and AvgPIC by locus after some individuals or populations have been deleted.*

Description

The locus metadata supplied by DArT has OneRatioRef, OneRatioSnp, PICRef, PICSnp, and AvgPIC included, but the allelic composition will change when some individuals,or populations, are removed from the dataset and so the initial statistics will no longer apply. This script recalculates these statistics and places the recalculated values in the appropriate place in the genlight object.

Usage

```
utils.recalc.avgpic(x, verbose = NULL)
```

Arguments

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

Details

If the locus metadata OneRatioRefISnp, PICRefISnp and/or AvgPIC do not exist, the script creates and populates them.

Value

The modified genlight object.

Author(s)

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

See Also

utils.recalc.metrics for recalculating all metrics, utils.recalc.callrate for recalculating CallRate, utils.recalc.freqhomref for recalculating frequency of homozygous reference, utils.recalc.freqhomsnp for recalculating frequency of homozygous alternate, utils.recalc.freqhet for recalculating frequency of heterozygotes, gl.recalc.maf for recalculating minor allele frequency, gl.recalc.rdepth for recalculating average read depth

Examples

```
#out <- utils.recalc.avgpic(testset.gl)
```

---

utils.recalc.callrate *A utility script to recalculate the callrate by locus after some populations have been deleted*

Description

SNP datasets generated by DArT have missing values primarily arising from failure to call a SNP because of a mutation at one or both of the restriction enzyme recognition sites. The locus metadata supplied by DArT has callrate included, but the call rate will change when some individuals are removed from the dataset. This script recalculates the callrate and places these recalculated values in the appropriate place in the genlight object. It sets the Call Rate flag to TRUE.

Usage

```
utils.recalc.callrate(x, verbose = NULL)
```

Arguments

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

Value

The modified genlight object

Author(s)

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

See Also

utils.recalc.metrics for recalculating all metrics, utils.recalc.avgpic for recalculating avgPIC, utils.recalc.freqhomref for recalculating frequency of homozygous reference, utils.recalc.freqhomsnp for recalculating frequency of homozygous alternate, utils.recalc.freqhet for recalculating frequency of heterozygotes, gl.recalc.maf for recalculating minor allele frequency, gl.recalc.rdepth for recalculating average read depth

Examples

```
#out <- utils.recalc.callrate(testset.gl)
```

---

utils.recalc.freqhets *A utility script to recalculate the frequency of the heterozygous SNPs by locus after some populations have been deleted*

Description

The locus metadata supplied by DArT has FreqHets included, but the frequency of the heterozygotes will change when some individuals are removed from the dataset.

Usage

```
utils.recalc.freqhets(x, verbose = NULL)
```

Arguments

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

Details

This script recalculates the FreqHets and places these recalculated values in the appropriate place in the genlight object.

Note that the frequency of the homozygote reference SNPS is calculated from the individuals that could be scored.

Value

The modified genlight object.

Author(s)

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

See Also

`utils.recalc.metrics` for recalculating all metrics, `utils.recalc.callrate` for recalculating CallRate, `utils.recalc.freqhomref` for recalculating frequency of homozygous reference, `utils.recalc.freqhomsnp` for recalculating frequency of homozygous alternate, `utils.recalc.AvgPIC` for recalculating RepAvg, `gl.recalc.maf` for recalculating minor allele frequency, `gl.recalc.rdepth` for recalculating average read depth

Examples

```
#out <- utils.recalc.freqhets(testset.gl)
```

---

utils.recalc.freqhomref

A utility script to recalculate the frequency of the homozygous reference SNP by locus after some populations have been deleted

---

Description

The locus metadata supplied by DArT has FreqHomRef included, but the frequency of the homozygous reference will change when some individuals are removed from the dataset.

Usage

```
utils.recalc.freqhomref(x, verbose = NULL)
```

Arguments

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

Details

This script recalculates the FreqHomRef and places these recalculated values in the appropriate place in the genlight object.

Note that the frequency of the homozygote reference SNPS is calculated from the individuals that could be scored.

Value

The modified genlight object

Author(s)

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

See Also

`utils.recalc.metrics` for recalculating all metrics, `utils.recalc.callrate` for recalculating CallRate, `utils.recalc.avgpic` for recalculating AvgPIC, `utils.recalc.freqhomsnp` for recalculating frequency of homozygous alternate, `utils.recalc.freqhet` for recalculating frequency of heterozygotes, `gl.recalc.maf` for recalculating minor allele frequency, `gl.recalc.rdepth` for recalculating average read depth

Examples

```
#result <- utils.recalc.freqhomref(testset.gl)
```

---

utils.recalc.freqhomsnp

A utility script to recalculate the frequency of the homozygous alternate SNP by locus after some populations have been deleted

---

Description

The locus metadata supplied by DArT has FreqHomSnp included, but the frequency of the homozygous alternate will change when some individuals are removed from the dataset.

Usage

```
utils.recalc.freqhomsnp(x, verbose = NULL)
```

Arguments

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

Details

This script recalculates the FreqHomSnp and places these recalculated values in the appropriate place in the genlight object.

Note that the frequency of the homozygote alternate SNPS is calculated from the individuals that could be scored.

Value

The modified genlight object.

Author(s)

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

See Also

`utils.recalc.metrics` for recalculating all metrics, `utils.recalc.callrate` for recalculating CallRate, `utils.recalc.freqhomref` for recalculating frequency of homozygous reference, `utils.recalc.avgpic` for recalculating AvgPIC, `utils.recalc.freqhet` for recalculating frequency of heterozygotes, `gl.recalc.maf` for recalculating minor allele frequency, `gl.recalc.rdepth` for recalculating average read depth

Examples

```
#out <- utils.recalc.freqhomsnp(testset.gl)
```

---

utils.recalc.maf *A utility script to recalculate the minor allele frequency by locus, typically after some populations have been deleted*

Description

The locus metadata supplied by DArT does not have MAF included, so it is calculated and added to the locus.metadata by this script. The minimum allele frequency will change when some individuals are removed from the dataset. This script recalculates the MAF and places these recalculated values in the appropriate place in the genlight object.

Usage

```
utils.recalc.maf(x, verbose = NULL)
```

Arguments

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

Value

The modified genlight dataset.

Author(s)

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

See Also

`utils.recalc.metrics` for recalculating all metrics, `utils.recalc.callrate` for recalculating CallRate, `utils.recalc.freqhomref` for recalculating frequency of homozygous reference, `utils.recalc.freqhomsnp` for recalculating frequency of homozygous alternate, `utils.recalc.freqhet` for recalculating frequency of heterozygotes, `gl.recalc.avgpic` for recalculating AvgPIC, `gl.recalc.rdepth` for recalculating average read depth

Examples

```
#f <- dartR::utils.recalc.maf(testset.gl)
```

---

utils.reset.flags *A utility script to reset to FALSE (or TRUE) the locus metric flags after some individuals or populations have been deleted.*

Description

The locus metadata supplied by DArT has OneRatioRef, OneRatioSnp, PICRef, PICSnp, and AvgPIC included, but the allelic composition will change when some individuals are removed from the dataset and so the initial statistics will no longer apply. This applies also to some variable calculated by dartR (e.g. maf). This script resets the locus metrics flags to FALSE to indicate that these statistics in the genlight object are no longer current. The verbosity default is also set, and in the case of SilcoDArT, the flags PIC and OneRatio are also set.

Usage

```
utils.reset.flags(x, set = FALSE, value = 2, verbose = NULL)
```

Arguments

x	Name of the genlight object containing the SNP data or tag presence/absence data (SilicoDArT) [required].
set	Set the flags to TRUE or FALSE [default FALSE].
value	Set the default verbosity for all functions, where verbosity is not specified [default 2].
verbose	Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default NULL].

Details

If the locus metrics do not exist then they are added to the genlight object but not populated. If the locus metrics flags do not exist, then they are added to the genlight object and set to FALSE (or TRUE).

Value

The modified genlight object

Author(s)

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

See Also

`utils.recalc.metrics` for recalculating all metrics, `utils.recalc.callrate` for recalculating CallRate, `utils.recalc.freqhomref` for recalculating frequency of homozygous reference, `utils.recalc.freqhomsnp` for recalculating frequency of homozygous alternate, `utils.recalc.freqhet` for recalculating frequency of heterozygotes, `gl.recalc.maf` for recalculating minor allele frequency, `gl.recalc.rdepth` for recalculating average read depth

Examples

```
#result <- utils.reset.flags(testset.gl)
```

---

utils.spautocor *Spatial autocorrelation coefficient calculations*

Description

Carries out calculation for spatial autocorrelation coefficient starting from a genetic and geogaphic distance matrix.

Usage

```
utils.spautocor(
  GD,
  GGD,
  permutation = FALSE,
  bootstrap = FALSE,
  bins = 10,
  reps
)
```

Arguments

GD	Genetic distance matrix.
GGD	Geographic distance matrix.
permutation	Whether permutation calculations for the null hypothesis of no spatial structure should be carried out [default TRUE].
bootstrap	Whether bootstrap calculations to compute the 95% confidence intervals around r should be carried out [default TRUE].
bins	The number of bins for the distance classes (i.e. `length(bins) == 1`) or a vectors with the break points. See details [default 5].
reps	The number to be used for permutation and bootstrap analyses [default 100].

Details

The code of this function is based one `spautocorr` from the package `PopGenReport`, which has been modified to fix a few bugs (as of `PopGenReport v 3.0.4` and allow calculations of bootstraps estimates.

See details from `gl.spatial.autoCorr` for a detailed explanation.

Value

Returns a data frame with the following columns:

1. Bin The distance classes
2. N The number of pairwise comparisons within each distance class
3. r.uc The uncorrected autocorrelation coefficient

if both `bootstap` and `permutation` are `FALSE` otherwise only `r` estimates are returned

Author(s)

Carlo Pacioni & Bernd Gruber

References

- Smouse PE, Peakall R. 1999. Spatial autocorrelation analysis of individual multiallele and multilocus genetic structure. Heredity 82: 561-573.
- Double, MC, et al. 2005. Dispersal, philopatry and infidelity: dissecting local genetic structure in superb fairy-wrens (Malurus cyaneus). Evolution 59, 625-635.
- Peakall, R, et al. 2003. Spatial autocorrelation analysis offers new insights into gene flow in the Australian bush rat, Rattus fuscipes. Evolution 57, 1182-1195.
- Smouse, PE, et al. 2008. A heterogeneity test for fine-scale genetic structure. Molecular Ecology 17, 3389-3400.
- Gonzales, E, et al. 2010. The impact of landscape disturbance on spatial genetic structure in the Guanacaste tree, Enterolobium cyclocarpum(Fabaceae). Journal of Heredity 101, 133-143.
- Beck, N, et al. 2008. Social constraint and an absence of sex-biased dispersal drive fine-scale genetic structure in white-winged choughs. Molecular Ecology 17, 4346-4358.

See Also

gl.spatial.autoCorr

Examples

```
# See gl.spatial.autoCorr
```

`utils.structure.evanno`

Util function for evanno plots

Description

These functions were copied from package strataG, which is no longer on CRAN (maintained by Eric Archer)

Usage

```
utils.structure.evanno(sr, plot = TRUE)
```

Arguments

`sr`	structure run object
`plot`	should the plots be returned

Author(s)

Bernd Gruber (bugs? Post to https://groups.google.com/d/forum/dartr); original implementation of Eric Archer https://github.com/EricArcher/strataG

`utils.structure.genind2gtypes`

structure util functions

Description

These functions were copied from package strataG, which is no longer on CRAN (maintained by Eric Archer)

Usage

```
utils.structure.genind2gtypes(x)
```

Arguments

`x`	a genind object

Value

a gtypes object

Author(s)

Bernd Gruber (bugs? Post to https://groups.google.com/d/forum/dartr); original implementation of Eric Archer https://github.com/EricArcher/strataG

`utils.structure.run` *Utility function to run Structure*

Description

These functions were copied from package strataG, which is no longer on CRAN (maintained by Eric Archer)

Usage

```
utils.structure.run(
  g,
  k.range = NULL,
  num.k.rep = 1,
  label = NULL,
  delete.files = TRUE,
  exec = "structure",
  ...
)
```

Arguments

`g`	a gtypes object [see `strataG`].
`k.range`	vector of values to for `maxpop` in multiple runs. If set to `NULL`, a single STRUCTURE run is conducted with `maxpops` groups. If specified, do not also specify `maxpops`.
`num.k.rep`	number of replicates for each value in `k.range`.
`label`	label to use for input and output files
`delete.files`	logical. Delete all files when STRUCTURE is finished?
`exec`	name of executable for STRUCTURE. Defaults to "structure".
`...`	arguments to be passed to `structureWrite`.

Value

`structureRun` a list where each element is a list with results from `structureRead` and a vector of the filenames used

`structureWrite` a vector of the filenames used by STRUCTURE

`structureRead` a list containing:

- `summary` new locus name, which is a combination of loci in group
- `q.mat` data.frame of assignment probabilities for each id
- `prior.anc` list of prior ancestry estimates for each individual where population priors were used
- `files` vector of input and output files used by STRUCTURE
- `label` label for the run

Author(s)

Bernd Gruber (bugs? Post to https://groups.google.com/d/forum/dartr); original implementation of Eric Archer https://github.com/EricArcher/strataG

---

zzz *Setting up the package*

Description

Setting theme, colors and verbosity

Usage

```
zzz
```

Format

An object of class `NULL` of length 0.

[,dartR,ANY,ANY,ANY-method *indexing dartR objects correctly...*

Description

indexing dartR objects correctly...

Usage

```
## S4 method for signature 'dartR,ANY,ANY,ANY'
x[i, j, ..., pop = NULL, treatOther = TRUE, quiet = TRUE, drop = FALSE]
```

Arguments

x	dartR object
i	index for individuals
j	index for loci
...	other parameters
pop	list of populations to be kept
treatOther	elements in other (and ind.metrics & loci.metrics) as indexed as well. default: TRUE
quiet	warnings are suppressed. default: TRUE
drop	reduced to a vector if a single individual/loci is selected. default: FALSE [should never set to TRUE]

Index

∗ **Exploration/visualisation functions**
gl.pcoa.plot, 109
gl.select.colors, 174
gl.select.shapes, 176
gl.smearplot, 189
∗ **Genetic variation within populations**
gl.test.heterozygosity, 195
∗ **data exploration functions**
gl.pcoa, 106
∗ **datasets**
bandicoot.gl, 6
platy, 226
possums.gl, 227
testset.gl, 228
testset.gs, 228
testset_metadata, 229
testset_pop_recode, 229
testset_SNPs_2Row, 229
zzz, 259
∗ **filter functions**
gl.filter.allna, 41
gl.filter.callrate, 42
gl.filter.heterozygosity, 46
gl.filter.hwe, 47
gl.filter.ld, 50
gl.filter.locmetric, 51
gl.filter.maf, 53
gl.filter.monomorphs, 55
gl.filter.overshoot, 56
gl.filter.pa, 57
gl.filter.parent.offspring, 58
gl.filter.rdepth, 60
gl.filter.reproducibility, 62
gl.filter.secondaries, 63
gl.filter.sexlinked, 64
gl.filter.taglength, 66
∗ **filters functions**
gl.filter.hamming, 44
∗ **inbreeding functions**
gl.grm, 72
gl.grm.network, 74
∗ **input data**
gl.read.dart, 124
∗ **ld functions**
gl.ld.distance, 88
gl.ld.haplotype, 90
∗ **reading functions**
gl.read.fasta, 125
∗ **reference genomes**
gl.blast, 16
∗ **report functions**
gl.report.bases, 134
gl.report.callrate, 136
gl.report.diversity, 138
gl.report.hamming, 140
gl.report.heterozygosity, 142
gl.report.hwe, 144
gl.report.ld.map, 149
gl.report.locmetric, 151
gl.report.maf, 154
gl.report.monomorphs, 156
gl.report.overshoot, 157
gl.report.pa, 158
gl.report.parent.offspring, 160
gl.report.rdepth, 163
gl.report.reproducibility, 164
gl.report.secondaries, 166
gl.report.sexlinked, 168
gl.report.taglength, 170
∗ **reporting functions**
gl.diagnostics.hwe, 25
∗ **simulation functions**
gl.sim.create_dispersal, 179
gl.sim.WF.run, 185
gl.sim.WF.table, 187
[,dartR,ANY,ANY,ANY-method, 259

A.mat, *72*
adegenet, *235*

bandicoot.gl, 6
basic.stats, *16*

cbind.dartR, 7

genind2genalex, *207*
gi2gl, 7
gl.alf, 8
gl.amova, 9
gl.assign.grm, 10
gl.assign.mahalanobis, 11
gl.assign.pa, 13
gl.assign.pca, *14*, 14
gl.basic.stats, 16
gl.blast, 16
gl.check.verbosity, 20
gl.collapse, 20
gl.compliance.check, 22
gl.costdistances, 23
gl.define.pop, 24
gl.diagnostics.hwe, 25
gl.diagnostics.sim, 27
gl.dist.ind, 29, *193*
gl.dist.pop, 30
gl.drop.ind, 32, *36*, *86*
gl.drop.loc, 33, *86*
gl.drop.pop, 34, *38*, *88*
gl.edit.recode.ind, 35
gl.edit.recode.pop, 37
gl.evanno, 38
gl.fdsim, 39
gl.filter.allna, *41*, 41, *44*, *46*, *50, 51*, *53*, *55–57*, *60, 61*, *63, 64*, *66, 67*, *83*, *148*
gl.filter.callrate, *29*, *42*, 42, *46*, *50, 51*, *53*, *55–57*, *60, 61*, *63, 64*, *66, 67*, *137*
gl.filter.hamming, 44, *141*
gl.filter.heterozygosity, *42*, *44*, 46, *50, 51*, *53*, *55–57*, *60, 61*, *63, 64*, *66, 67*, *144*
gl.filter.hwe, *42*, *44*, *46*, 47, *51*, *53*, *55–57*, *60, 61*, *63, 64*, *66, 67*, *148*
gl.filter.ld, *42*, *44*, *46*, *50*, 50, *53*, *55–57*, *60, 61*, *63, 64*, *66, 67*, *151*
gl.filter.locmetric, *42*, *44*, *46*, *50, 51*, 51, *55–57*, *60, 61*, *63, 64*, *66, 67*, *151–153*
gl.filter.maf, *42*, *44*, *46*, *50, 51*, *53*, 53, *55–57*, *60, 61*, *63, 64*, *66, 67*, *154, 155*
gl.filter.monomorphs, *42*, *44*, *46*, *50, 51*, *53*, *55*, 55, *56, 57*, *60, 61*, *63, 64*, *66, 67*, *131–133*, *156*
gl.filter.overshoot, *42*, *44*, *46*, *50, 51*, *53*, *55*, 56, *57*, *60, 61*, *63, 64*, *66, 67*, *158*
gl.filter.pa, *42*, *44*, *46*, *50, 51*, *53*, *55, 56*, 57, *60, 61*, *63, 64*, *66, 67*
gl.filter.parent.offspring, *42*, *44*, *46*, *50, 51*, *53*, *55–57*, 58, *61*, *63, 64*, *66, 67*, *162*
gl.filter.rdepth, *42*, *44*, *46*, *50, 51*, *53*, *55–57*, *60*, 60, *61*, *63, 64*, *66, 67*, *163, 164*
gl.filter.reproducibility, *42*, *44*, *46*, *50, 51*, *53*, *55–57*, *60, 61*, 62, *64*, *66, 67*, *165*
gl.filter.secondaries, *42*, *44*, *46*, *50, 51*, *53*, *55–57*, *60, 61*, *63*, 63, *66, 67*, *167, 168*, *243*
gl.filter.sexlinked, *42*, *44*, *46*, *50, 51*, *53*, *55–57*, *60, 61*, *63, 64*, 64, *67*
gl.filter.taglength, *42*, *44*, *46*, *50, 51*, *53*, *55–57*, *60, 61*, *63, 64*, *66*, 66, *171*
gl.fixed.diff, 67, *226*
gl.fst.pop, 69
gl.genleastcost, 70
gl.grm, 72, *74*, *76*, *193*
gl.grm.network, *73*, 74
gl.He, 77
gl.Ho, 77
gl.hwe.pop, 78
gl.ibd, 79
gl.impute, 82
gl.install.vanilla.dartR, 83
gl.join, 84
gl.keep.ind, *33*, *36*, 85
gl.keep.loc, *34*, 86
gl.keep.pop, *35*, *38*, 87
gl.ld.distance, 88, *91*
gl.ld.haplotype, *89*, 90
gl.LDNe, 92
gl.list.reports, *19*, *59, 60*, 94, *108*, *118*, *120*, *139*, *153*, *155*, *160*, *162*, *171*, *193*, *196*
gl.load, 94, *174*
gl.make.recode.ind, 95
gl.make.recode.pop, 96
gl.map.interactive, 97

gl.map.structure, 99
gl.merge.pop, *38*, 101
gl.nhybrids, 102
gl.outflank, 104
gl.pcoa, 106, *111*
gl.pcoa.plot, *109*, 109, *175*, *177*, *190*
gl.percent.freq, 112
gl.play.history, 113
gl.plot.heatmap, 114
gl.plot.network, 115
gl.plot.structure, *99–101*, 117
gl.print.history, *19*, 119
gl.print.reports, *19*, *59, 60*, *94*, *108*, *118*, 120, *139*, *153*, *155*, *160*, *162*, *171*, *193*, *196*
gl.propShared, 121, *193*
gl.random.snp, 121
gl.read.csv, 122
gl.read.dart, *34*, 124, *128*
gl.read.fasta, 125
gl.read.silicodart, 127
gl.read.vcf, 128
gl.reassign.pop, *38*, 129
gl.recalc.metrics, 130, *132*
gl.recode.ind, *36*, 131
gl.recode.pop, *38*, *132*, 132, *133*
gl.rename.pop, 133
gl.report.bases, 134, *137*, *139*, *141*, *144*, *148*, *151*, *153*, *155, 156*, *158*, *160*, *162*, *164, 165*, *168*, *170, 171*
gl.report.callrate, *44*, *135*, 136, *139*, *141*, *144*, *148*, *151*, *153*, *155, 156*, *158*, *160*, *162*, *164, 165*, *168*, *170, 171*
gl.report.diversity, *135*, *137*, 138, *141*, *144*, *148*, *151*, *153*, *155, 156*, *158*, *160*, *162*, *164, 165*, *168*, *170, 171*
gl.report.hamming, *135*, *137*, *139*, 140, *144*, *148*, *151*, *153*, *155, 156*, *158*, *160*, *162*, *164, 165*, *168*, *170, 171*
gl.report.heterozygosity, *135*, *137*, *139*, *141*, 142, *148*, *151*, *153*, *155, 156*, *158*, *160*, *162*, *164, 165*, *167, 168*, *170, 171*, *218*, *243*
gl.report.hwe, *26, 27*, *50*, *135*, *137*, *139*, *141*, *144*, 144, *151*, *153*, *155, 156*, *158*, *160*, *162*, *164, 165*, *168*, *170, 171*
gl.report.ld, 148
gl.report.ld.map, *50, 51*, *88*, *135*, *137*, *139*, *141*, *144*, *148*, 149, *151*, *153*, *155, 156*, *158*, *160*, *162*, *164, 165*, *168*, *170, 171*
gl.report.locmetric, *135*, *137*, *139*, *141*, *144*, *148*, *151*, 151, *155, 156*, *158*, *160*, *162*, *164, 165*, *168*, *170, 171*
gl.report.maf, *135*, *137*, *139*, *141*, *144*, *148*, *151*, *153*, 154, *156*, *158*, *160*, *162*, *164, 165*, *168*, *170, 171*
gl.report.monomorphs, *135*, *137*, *139*, *141*, *144*, *148*, *151*, *153*, *155*, 156, *158*, *160*, *162*, *164, 165*, *168*, *170, 171*
gl.report.overshoot, *135*, *137*, *139*, *141*, *144*, *148*, *151*, *153*, *155, 156*, 157, *160*, *162*, *164, 165*, *168*, *170, 171*
gl.report.pa, *135*, *137*, *139*, *141*, *144*, *148*, *151*, *153*, *155, 156*, *158*, 158, *162*, *164, 165*, *168*, *170, 171*
gl.report.parent.offspring, *58–60*, *135*, *137*, *139*, *141*, *144*, *148*, *151*, *153*, *155, 156*, *158*, *160*, 160, *164, 165*, *168*, *170, 171*
gl.report.rdepth, *59, 60*, *135*, *137*, *139*, *141*, *144*, *148*, *151*, *153*, *155, 156*, *158*, *160*, *162*, 163, *165*, *168*, *170, 171*
gl.report.reproducibility, *59, 60*, *63*, *135*, *137*, *139*, *141*, *144*, *148*, *151*, *153*, *155, 156*, *158*, *160*, *162*, *164*, 164, *168*, *170, 171*
gl.report.secondaries, *135*, *137*, *139*, *141*, *143, 144*, *148*, *151*, *153*, *155, 156*, *158*, *160*, *162*, *164, 165*, 166, *170, 171*, *218*
gl.report.sexlinked, *135*, *137*, *139*, *141*, *144*, *148*, *151*, *153*, *155, 156*, *158*, *160*, *162*, *164, 165*, *168*, 168, *171*
gl.report.taglength, *135*, *137*, *139*, *141*, *144*, *148*, *151*, *153*, *155, 156*, *158*, *160*, *162*, *164, 165*, *168*, *170*, 170
gl.run.structure, *38, 39*, *99–101*, *117*, 172
gl.save, *95*, 173
gl.select.colors, *111*, 174, *177*, *190*
gl.select.shapes, *111*, *175*, 176, *190*
gl.set.verbosity, 177
gl.sfs, 178
gl.sim.create_dispersal, 179, *185*, *187*,

189
gl.sim.emigration, 181
gl.sim.ind, 182
gl.sim.mutate, 183
gl.sim.offspring, 184
gl.sim.WF.run, *28*, *179*, *180*, 185, *187*, *189*
gl.sim.WF.table, *180*, *185*, *187*, 187
gl.smearplot, *111*, *175*, *177*, 189
gl.spatial.autoCorr, 190, *257*
gl.subsample.loci, 194
gl.test.heterozygosity, 195
gl.tree.nj, 197
gl.write.csv, 198
gl2bayescan, 199
gl2bpp, 200
gl2demerelate, 201
gl2eigenstrat, 202
gl2fasta, 203
gl2faststructure, 205
gl2gds, 206
gl2genalex, 207
gl2genepop, 208
gl2geno, 209
gl2gi, *149*, 210
gl2hiphop, 211
gl2phylip, 212
gl2plink, 213
gl2related, 215
gl2sa, 216
gl2sfs, 217
gl2shp, 218
gl2snapp, 219
gl2structure, 220
gl2svdquartets, 221
gl2treemix, 222
gl2vcf, 223

heatmap.2, *114*
HWChisq, *48*, *146*
HWExactPrevious, *79*
HWExactStats, *48*, *78*, *146*
HWPerm, *79*
HWTernaryPlot, *147*

interactive_reference, 224
interactive_sim_run, 225
is.fixed, *69*, 225

landgenreport, *72*
layout_with_fr, *76*
layout_with_graphopt, *76*
layout_with_kk, *76*
layout_with_mds, *76*
ld, *91*, *149*, *150*
lgrMMRR, *70*, *72*

mantel, *81*

p.adjust, *49*, *146*
platy, 226
popgenreport, *72*
possums.gl, 227
providers, *100*

rbind.dartR, 227
rSPDistance, *71*

scale_fill_viridis, *91*
snpgdsClose, *207*
stamppFst, *80*, *81*
stamppNeisD, *80*, *81*

testset.gl, 228
testset.gs, 228
testset_metadata, 229
testset_pop_recode, 229
testset_SNPs_2Row, 229
theme_dartR, 230

utils.assignment, 230
utils.assignment_2, 231
utils.assignment_3, 232
utils.assignment_4, 233
utils.basic.stats, 234
utils.check.datatype, 234
utils.dart2genlight, 235
utils.dist.binary, 236
utils.dist.ind.snp, 238
utils.flag.start, 239
utils.hamming, *45*, *141*, 240
utils.het.pop, 241
utils.jackknife, 241
utils.n.var.invariant, *168*, *218*, 243
utils.outflank, *105*, 244
utils.outflank.MakeDiploidFSTMat, *105*, 246
utils.outflank.plotter, *105*, 246
utils.read.dart, *125*, 247
utils.recalc.avgpic, 248

utils.recalc.callrate, 249
utils.recalc.freqhets, 250
utils.recalc.freqhomref, 251
utils.recalc.freqhomsnp, 252
utils.recalc.maf, 253
utils.reset.flags, 254
utils.spautocor, 255
utils.structure.evanno, 257
utils.structure.genind2gtypes, 257
utils.structure.run, 258

wassermann, *70*, *72*

zzz, 259