Package ‘dartR.popgen’

November 21, 2023

Type Package

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

Version 0.32

Date 2023-11-21

Description Facilitates the analysis of SNP (single nucleotide polymorphism) and silicodart (presence/absence) data. ‘dartR.popgen’ provides a suit of functions to analyse such data in a population genetics context. It provides several functions to calculate population genetic metrics and to study population structure. Quite a few functions need additional software to be able to run (gl.run.structure(), gl.blast(), gl.LDNe()). You find detailed description in the help pages how to download and link the packages so the function can run the software. ‘dartR.popgen’ is part of the the ‘dartRverse’ suit of packages. Gruber et al. (2018) <doi:10.1111/1755-0998.12745>. Mijangos et al. (2022) <doi:10.1111/2041-210X.13918>.

Encoding UTF-8

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

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

Suggests SIBER, expm, fields, gplots, gridExtra, igraph, iterpc, label.switching, leaflet,plyr, proxy, purrr, qvalue, raster, reshape2, scales, snpStats, tidyverse, viridis, zoo, gsubfn, sp

License GPL (>= 3)

RoxygenNote 7.2.3

NeedsCompilation no

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

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 2023-11-21 18:10:02 UTC

R topics documented:

```
gl.blast ......................................................... 2
gl.collapse ..................................................... 6
gl.evanno ....................................................... 7
gl.ld.distance .................................................... 8
gl.ld.haplotype .................................................. 10
gl.LDNe .......................................................... 12
gl.map.structure .................................................. 14
gl.nhybrids ...................................................... 16
gl.outflank ....................................................... 18
gl.plot.faststructure ........................................... 20
gl.plot.structure ................................................ 22
gl.run.faststructure ............................................. 24
gl.run.structure ................................................ 26
gl.sfs ............................................................. 28
utils.outflank ................................................... 29
utils.outflank.MakeDiploidFSTMat ................................ 31
utils.outflank.plotter .......................................... 32
utils.structure.evanno ......................................... 33
utils.structure.genind2gtypes .................................. 33
utils.structure.run ............................................. 34```

Index 36

---

**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
gl.blast

your private collection, contigs, scaffolds or any other genetic sequence that you would like to use as reference.

Usage

```r
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
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 working directory (plot.dir tempdir if not set).

**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**


**See Also**

`gl.print.history`

**Examples**

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

## End(Not run)
```
gl.collapse  

Collapses a distance matrix by amalgamating populations with pair-wise fixed difference count less than 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

```r
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

```r
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.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**

```r
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](https://groups.google.com/d/forum/dartr))
References


See Also

gl.run.structure, clumpp,

Examples

# examples need structure to be installed on the system (see above)
## 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)
head(qmat)
qmat.map.structure(qmat, bc, K=3, scalex=1, scaley=0.5)
## End(Not run)

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

```r
cl.ld.distance(
  ld_report,
  ld_resolution = 1e+05,
  pop_colors = NULL,
  plot_theme = NULL,
  plot.out = TRUE,
  plot.file = NULL,
  plot.dir = NULL,
  verbose = NULL
)
```
Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ld_report</td>
<td>Output from function <code>gl.report.ld.map</code> [required].</td>
</tr>
<tr>
<td>ld_resolution</td>
<td>Resolution at which LD should be reported in number of base pairs [default NULL].</td>
</tr>
<tr>
<td>pop_colors</td>
<td>A color palette for box plots by population or a list with as many colors as there are populations in the dataset [default NULL].</td>
</tr>
<tr>
<td>plot_theme</td>
<td>User specified theme [default NULL].</td>
</tr>
<tr>
<td>plot.out</td>
<td>Specify if plot is to be produced [default TRUE].</td>
</tr>
<tr>
<td>plot.file</td>
<td>Name for the RDS binary file to save (base name only, exclude extension) [default NULL].</td>
</tr>
<tr>
<td>plot.dir</td>
<td>Directory in which to save files [default = working directory]</td>
</tr>
<tr>
<td>verbose</td>
<td>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].</td>
</tr>
</tbody>
</table>

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


See Also

Other ld functions: `gl.ld.haplotype()`

Examples

```r
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 = 1000000)
  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**

```r
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,
    plot.file = NULL,
    plot.dir = NULL,
    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 analysed [default NULL].
- **chrom_name**  
  Name of the chromosome to analyse. If NULL all the chromosomes are analysed [default NULL].
- **ld_max_pairwise**  
  Maximum distance in number of base pairs at which LD should be calculated [default 10000000].
**gl.ld.haplotype**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maf</td>
<td>Minor allele frequency (by population) threshold to filter out loci. If a value &gt; 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].</td>
</tr>
<tr>
<td>ld_stat</td>
<td>The LD measure to be calculated: &quot;LLR&quot;, &quot;OR&quot;, &quot;Q&quot;, &quot;Covar&quot;, &quot;D.prime&quot;, &quot;R.squared&quot;, and &quot;R&quot;. See ld (package snpStats) for details [default &quot;R.squared&quot;].</td>
</tr>
<tr>
<td>ind.limit</td>
<td>Minimum number of individuals that a population should contain to take it into account to report loci in LD [default 10].</td>
</tr>
<tr>
<td>min_snps</td>
<td>Minimum number of SNPs that should have a haplotype to call it [default 10].</td>
</tr>
<tr>
<td>ld_threshold_haplo</td>
<td>Minimum LD between adjacent SNPs to call a haplotype [default 0.5].</td>
</tr>
<tr>
<td>coordinates</td>
<td>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].</td>
</tr>
<tr>
<td>color_haplo</td>
<td>Color palette for haplotype plot. See details [default &quot;viridis&quot;].</td>
</tr>
<tr>
<td>color_het</td>
<td>Color for heterozygosity [default &quot;deeppink&quot;].</td>
</tr>
<tr>
<td>plot.out</td>
<td>Specify if heatmap plot is to be produced [default TRUE].</td>
</tr>
<tr>
<td>plot.file</td>
<td>Name for the RDS binary file to save (base name only, exclude extension) [default NULL] temporary directory (tempdir) [default FALSE].</td>
</tr>
<tr>
<td>plot.dir</td>
<td>Directory in which to save files [default = working directory]</td>
</tr>
<tr>
<td>verbose</td>
<td>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].</td>
</tr>
</tbody>
</table>

**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

```r
require("dartR.data")
x <- platypus.gl
dx <- gl.filter.callrate(x, threshold = 1)
# only the first 20 individuals because of speed during tests
x <- gl.keep.pop(x, pop.list = "TENTERFIELD"[1:20, ])
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 = 1000000)
```

---

**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/](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**

```r
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 = gl.select.colors(x, verbose = 0),
  plot.file = NULL,
  plot.dir = NULL,
  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 executa-
bles 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, there-
fore 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  population colors with as many colors as there are populations in the dataset
[default discrete_palette].
plot.file  Name for the RDS binary file to save (base name only, exclude extension) [de-
fault NULL] temporary directory (tempdir) [default FALSE].
plot.dir  Directory in which to save files [default = working directory]
verbose  Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress
and results summary; 5, full report [default 2, 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", outputpath = tempdir(),
   neest.path = "/path_to Ne-21",
   critical = c(0, 0.05), singleton.rm = TRUE, mating = "random"
)
nes

## End(Not run)
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](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](https://groups.google.com/d/forum/dartr))

References


See Also

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

Examples

```r
# examples need structure to be installed on the system (see above)
## 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
t <- gl.plot.structure(sr, k=2:4)
qmat <- gl.plot.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.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 NewHybrids.

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/software/new_hybs_doc1_1Beta3.pdf

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

```r
.gl.nhybrids(
  gl,
  outpath = tempdir(),
  p0 = NULL,
  p1 = NULL,
  threshold = 0,
  method = "random",
  plot = TRUE,
  plot_theme = theme_dartR(),
  plot_colors = gl.select.colors(ncolors = 2, verbose = 0),
  pprob = 0.95,
)```

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) to select 200 loci for NewHybrids [default random]. Previous AvgPic does not work anymore!
- **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


Examples

```r
## 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

```r
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

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 OutFLANK from a genlight object)

See Also
utils.outflank, utils.outflank.plotter, utils.outflank.MakeDiploidFSTMat

Examples

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

Plots fastStructure analysis results (Q-matrix)

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

Usage
```r
gl.plot.faststructure(
  sr,  # fastStructure run object from `gl.run.faststructure` [required].
  k.range,  # 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 = "greedyLargeK",  # The algorithm to use to infer the correct permutations. One of 'greedy' or 'greedyLargeK' or 'stephens' [default "greedyLargeK"].
  iter_clumpp = 100,  # The number of iterations to use if running either 'greedy' 'greedyLargeK' [default 100].
  clumpak = TRUE,  # Whether use the Clumpak method (see details) [default TRUE].
  plot_theme = NULL,  # Theme for the plot. See Details for options [default NULL].
  colors_clusters = NULL,  # A color palette for clusters (K) or a list with as many colors as there are clusters (K) [default NULL].
  ind_name = TRUE,  # Whether to plot individual names [default TRUE].
  border_ind = 0.15  # The width of the border line between individuals [default 0.25].
)
```

Arguments
- **sr**: fastStructure run object from `gl.run.faststructure` [required].
- **k.range**: 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].

Details
The function outputs a barplot which is the typical output of fastStructure.
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.

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

- 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


See Also

gl.run.faststructure

Examples

```r
## Not run:
t1 <- gl.filter.callrate(platypus.gl, threshold = 1)
res <- gl.run.faststructure(t1,
    exec = "./fastStructure", k.range = 2:3,
    num.k.rep = 2, output = paste0(getwd(), "/res_str")
)
qmat <- gl.plot.faststructure(res, k.range = 2:3)
gl.map.structure(qmat, K = 2, t1, scalex = 1, scaley = 0.5)

## End(Not run)
```
**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**

```r
gl.plot.structure(
  sr,
  K = NULL,
  met_clumpp = "greedyLargeK",
  iter_clumpp = 100,
  clumpak = TRUE,
  plot_theme = NULL,
  color_clusters = NULL,
  ind_name = TRUE,
  border_ind = 0.15,
  plot.out = TRUE,
  plot.file = NULL,
  plot.dir = NULL,
  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].
- `color_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].
plot.file Name for the RDS binary file to save (base name only, exclude extension) [default NULL]
plot.dir Directory in which to save files [default = working directory]
verbose Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default NULL, unless specified using gl.set.verbosity]

Details

The function outputs a barplot which is the typical output of structure. For an 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 working directory specified in plot.dir (tempdir) if plot.file is set.

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


• 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

```r
# examples need structure to be installed on the system (see above)
## 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
qumat <- gl.plot.structure(sr, K=3)
head(qmat)
gl.map.structure(qmat, K=3, bc, scalex=1, scaley=0.5)
## End(Not run)
```

---

**gl.run.faststructure**  
Runs a faststructure analysis using a genlight object

Description

This function takes a genlight object and runs a faststructure analysis.

Usage

```r
gl.run.faststructure(
  x,
  k.range,
  num.k.rep,
  exec = './fastStructure',
  output = getwd(),
  tol = 1e-05,
  prior = "simple",
  cv = 0,
  seed = NULL
)
```

Arguments

- **x**: Name of the genlight object containing the SNP data [required].
- **k.range**: Range of the number of populations [required].
- **num.k.rep**: Number of replicates [required].
- **exec**: Full path and name+extension where the fastStructure executable is located [default working directory "./fastStructure"].
- **output**: Path to output file [default getwd()].
- **tol**: Convergence criterion [default 10e-6].
- **prior**: Choice of prior: simple or logistic [default "simple"].
- **cv**: Number of test sets for cross-validation, 0 implies no CV step [default 0].
- **seed**: Seed for random number generator [default NULL].
Details

Download faststructure binary for your system from here (only runs on Mac or Linux):
https://github.com/StuntsPT/Structure_threader/tree/master/structure_threader/bins

Move faststructure file to working directory. Make file executable using terminal app.

```r
system(paste0("chmod u+x ",getwd(), "/faststructure"))
```

Download plink binary for your system from here:
https://www.cog-genomics.org/plink/

Move plink file to working directory. Make file executable using terminal app.

```r
system(paste0("chmod u+x ",getwd(), "/plink"))
```

To install fastStructure dependencies follow these directions: https://github.com/rajanil/fastStructure

fastStructure performs inference for the simplest, independent-loci, admixture model, with two choices of priors that can be specified using the –prior parameter. Thus, unlike Structure, fastStructure does not require the mainparams and extraparam files. The inference algorithm used by fastStructure is fundamentally different from that of Structure and requires the setting of far fewer options.

To identify the number of populations that best approximates the marginal likelihood of the data, the marginal likelihood is extracted from each run of K, averaged across replications and plotted.

Value

A list in which each list entry is a single faststructure run output (there are k.range * num.k.rep number of runs).

Author(s)

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

References


Examples

```r
## Not run:
# Please note: faststructure needs to be installed
# Please note: faststructure is not available for windows
t1 <- gl.filter.callrate(platypus.gl, threshold = 1)
res <- gl.run.faststructure(t1,
   exec = "./fastStructure", k.range = 2:3,
   num.k.rep = 2, output = paste0(getwd(), "/res_str")
)
qmat <- gl.plot.faststructure(res, k.range = 2:3)
gl.map.structure(qmat, K = 2, t1, scalex = 1, scaley = 0.5)

## End(Not run)
```
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(),
  plot.dir = NULL,
  plot.file = NULL,
  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()].

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

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

verbose  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 your system (non GUI version) from here STRUCTURE.

Format note

For this function to work, make sure that individual and population names have no spaces. To substitute spaces by underscores you could use the R function gsub as below.

```r
popNames(gl) <- gsub(" ", "_", popNames(gl)); indNames(gl) <- gsub(" ", "_", indNames(gl))
```

It's also worth noting that Structure truncates individual names at 11 characters. The function will fail if the names of individuals are not unique after truncation. To avoid this possible problem, a number sequence, as shown in the code below, might be used instead of individual names.

```r
indNames(gl) <- as.character(1:length(indNames(gl)))
```

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 (clump) 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


Examples

```r
# examples need structure to be installed on the system (see above)
## 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)
head(qmat)
gl.map.structure(qmat, bc, scalex=1, scaley=0.5)
## End(Not run)
```
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

```r
gl.sfs(
  x,
  minbinsize = 0,
  folded = TRUE,
  singlepop = FALSE,
  plot.out = TRUE,
  plot.file = NULL,
  plot.dir = NULL,
  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].
- **plot.file**: Name for the RDS binary file to save (base name only, exclude extension) [default NULL]
- **plot.dir**: Directory in which to save files [default = working directory]
- **verbose**: Verbosity: 0, silent or fatal errors; 1, begin and end; 2, progress log; 3, progress and results summary; 5, full report [default 2, 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


Examples

gl.sfs(bandicoot.gl, singlepop = TRUE)
gl.sfs(possums.gl[c(1:5, 31:33), ], minbinsize = 1)

---

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 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].

H\text{\textsubscript{min}}
The minimum heterozygosity required before including calculations from a locus [default 0.1].

NumberOfSamples
The number of spatial locations included in the data set.

q\text{\textsubscript{threshold}}
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
• FSTNoCorrb: 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**

```r
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 distribution (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.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**

```r
utils.structure.evanno(sr, plot = TRUE)
```

**Arguments**

- `sr` : structure run object
- `plot` : should the plots be returned

**Value**

returns a list of dataframes (structure results) and a list of plots

**Author(s)**

Bernd Gruber (bugs? Post to [https://groups.google.com/d/forum/dartr](https://groups.google.com/d/forum/dartr)); original implementation of Eric Archer [https://github.com/EricArcher/strataG](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**

```r
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

Usage

```r
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.

Description

These functions were copied from package strataG, which is no longer on CRAN (maintained by Eric Archer)
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
Index

* ld functions
  - gl.ld.distance, 8
  - gl.ld.haplotype, 10
* reference genomes
  - gl.blast, 2
  - gl.collapse, 6
  - gl.evanno, 7
  - gl.ld.distance, 8, 11
  - gl.ld.haplotype, 9, 10
  - gl.LDNe, 12
  - gl.map.structure, 14
  - gl.nhybrids, 16
  - gl.outflank, 18
  - gl.plot.faststructure, 20
  - gl.plot.structure, 14, 15, 22
  - gl.print.history, 5
  - gl.report.ld.map, 9
  - gl.run.faststructure, 20, 24
  - gl.run.structure, 7, 8, 14, 15, 22, 26
  - gl.sfs, 28

ld, 11

providers, 14

scale_fill_viridis, 11

utils.outflank, 19, 29
utils.outflank.MakeDiploidFSTMat, 19, 31
utils.outflank.plotter, 19, 32
utils.structure.evanno, 33
utils.structure.genind2gtypes, 33
utils.structure.run, 34