Package 'sGD'

March 6, 2018

Type Package

Title Spatially explicit estimation of genetic diversity indices and Wright's neighborhood size (NS)

Version 2.1

Date 2015-08-06

Author Andrew Shirk

Maintainer Andrew Shirk <ashirk@uw.edu>

BugReports https://github.com/Andrew-Shirk/sGD/issues

Description sGD provides spatially explicit estimates of genetic diversity indices and Wright's neighborhood size in continuous populations isolated by distance or resistance as described in Shirk & Cushman 2014. Inputs include a set of spatially referenced codominant marker genotypes, a matrix of pairwise distances between all sampled individuals (either in Euclidean units or as effective distances quantified by methods such as least-cost-path or circuit theory), and several parameters, including the minimum neighborhood size and the neighborhood radius (specified in Euclidean or effective distances). sGD defines a local neighborhood extent (based on a user-specified radius) around each sample location, and then calculates genetic diversity indices and/or Wright's neighborhood size (a local measure of effective population size that accounts for the continuous structure of populations isolated by distance). Requires the NeEstimator program (version 2.01) available at http://molecularfisherieslaboratory.com.au/neestimator-software.

Depends R (>= 3.0.0)

Imports adegenet (>= 2.0.0),
hierfstat,
raster,
sp,
ecodist,
gdistance,

Suggests knitr,
ggplot2,
maptools

License GPL (>= 2)

VignetteBuilder knitr

RoxygenNote 6.0.1

R topics documented:

`distmat` *Calculate a pairwise landscape distance matrix (Euclidian or cost-distance).*

Description

Calculate a pairwise landscape distance matrix (Euclidian or cost-distance).

Usage

```
distmat(sp_points, method, file_name = NULL, landscape = NULL)
```

Arguments

`sp_points`	An object of class SpatialPoints (see sp package for details).
`method`	Specify the type of distance matrix to be produced, using "ed" for Euclidean distance and "cd" for cost-weighted (i.e. effective) distance. Accurate distance calculations require a projected coordinate system (e.g. UTM), so do not use geographical coordinates (e.g. longlat). If you calculate cost-weighted distances, make sure the `points` projection is the same as the `landscape` raster.
`file_name`	(optional) A character string that will be appended to the beginning of the output filename. If no name is specified, no file will be written to the working directory.
`landscape`	An object of class RasterLayer (see raster package for details)

Value

An NxN (N= sample size: i.e. nrow(xy)) matrix of pairwise Euclidean and/or effective landscape distances written to .csv comma delimited files with edmat or cdmat appended to the end of the `filename`.

Examples

```
library(sGD)
library(raster)
library(sp)

# read in locations and landscape data
xy_file <- system.file("extdata","sGD_demo_xy.csv",package="sGD")
landscape_ascii <- system.file("extdata","sGD_demo_IBR_landscape.asc",package="sGD")

# convert locations to SpatialPoints (sp package)
proj <- "+proj=utm +zone=10 +datum=NAD83"
sp_points <- SpatialPoints(read.csv(xy_file)[,c(2,3)],proj4string=CRS(proj))

# convert landscape_ascii to raster object (raster package)
```

```
landscape <- raster(landscape_ascii,crs=CRS(proj))

# specify output file_name
file_name <- "sGD_demo"

# run ed and cd matrix calculations
ed <- distmat(sp_points,method="ed",file_name = file_name)
cd <- distmat(sp_points,method="cd",file_name = file_name,landscape=landscape)
```

`infer2sigma` *Infer the radius of Wright's genetic neighborhood from codominant marker genotypes. The correct radius is equal to 2 sigma, where sigma is the mean parent-offspring dispersal distance.*

Description

Infer the radius of Wright's genetic neighborhood from codominant marker genotypes. The correct radius is equal to 2 sigma, where sigma is the mean parent-offspring dispersal distance.

Usage

```
infer2sigma(genind_obj, xy, dist.mat, distances, min_N, max_N = NULL,
  min_distances = 1, max_absFIS = NULL, raw_out = FALSE)
```

Arguments

`genind_obj`	A genind object (created by the adegenet package function import2genind or related methods) containing individual genotypes. The order of the individuals must be the same as the order in the xy and dist.mat inputs below.
`xy`	A dataframe containing 3 columns in the following order: individual IDs, X coordinates, and Y coordinates. The order of the rows must match the order in the genind_obj and dist.mat inputs.
`dist.mat`	An NxN (N= sample size) matrix of pairwise landscape distances (Euclidean or effective). The `distmat` function in the sGD package may be used to produce Euclidean and cost-weighted distance matrices. The order of the rows and columns in the matrix must match the order in the xy and genind_obj inputs.
`distances`	A vector if distances at which to evaluate the evidence for the correct neighborhood radius.
`min_N`	The minimum sample size per neighborhood for indices to be calculated. NA is returned for neighborhoods < `min_N`.
`max_N`	Optional. The maximum sample size per neighborhood for indices to be calculated. If the number of individuals in the neighborhood exceeds `max_N`, a sample of size `max_N` will be used from the neighborhood to compute the metrics and output files specified by the user. Note that if `max_N` is specified, and the value is too small to be representative of the neighobrhood, the results could differ significantly compared to if all individuals in the neighborhood were used.
`min_distances`	The minimum number of distances allowed.
`max_absFIS`	The maximum absolute FIS value allowed. If NULL, all FIS values are considered valid.
`raw_out`	Boolean. Do you want to write the neighborhood FIS values for each radius to a csv file?

Examples

```
library(sGD)
library(adegenet)

# read in genotypes, locations, and distance matrix
genepop.file <- system.file("extdata","sGD_demo_IBR.gen",package="sGD")
xy = read.csv(system.file("extdata","sGD_demo_xy.csv",package="sGD"))
dist.mat <- as.matrix(read.csv(system.file("extdata","sGD_demo_cdmat.csv",package="sGD"),
                               header=FALSE))

# convert genepop to genind (make sure you specify the correct allele code digits - ncode)
genind_obj <- read.genepop(genepop.file,ncode=3L,quiet=TRUE)

# specify distances to evaluate
distances = c(8000,12000,16000,20000,24000)

# run infer2sigma
est2sigma <- infer2sigma(genind_obj,xy,dist.mat,distances,min_N=20)

NH_radii = est2sigma$opt_radius
NH_radii[which(est2sigma$num_distances < 4)] = NA
NH_radii[which(abs(est2sigma$FIS_error) > 0.02)] = NA
```

---

sGD *Calculate spatially explicit indicies of genetic diversity and Wright's neighborhood size (NS).*

Description

Calculate spatially explicit indicies of genetic diversity and Wright's neighborhood size (NS).

Usage

```
sGD(genind_obj, xy, dist.mat, NH_radius, min_N, max_N = NULL,
  metrics = NULL, NHmat_ans = FALSE, genout_ans = FALSE,
  file_name = NULL, NeEstimator_dir = NULL)
```

Arguments

genind_obj	A genind object (created by the adegenet package function import2genind or related methods) containing individual genotypes. The order of the individuals must be the same as the order in the xy and dist.mat inputs below.
xy	A dataframe containing 3 columns in the following order: individual IDs, X coordinates, and Y coordinates. The order of the rows must match the order in the genind_obj and dist.mat inputs.
dist.mat	An NxN (N= sample size) matrix of pairwise landscape distances (Euclidean or effective). The distmat function in the sGD package may be used to produce Euclidean and cost-weighted distance matrices. The order of the rows and columns in the matrix must match the order in the xy and genind_obj inputs.

`NH_radius` A single value to be used for all neighborhoods, or a vector of genetic neighborhood radii, optionally obtained from using the `infer2sigma` function. Note that if you specify a vector of radii, sGD will not calculate metrics when the radius value is NA.

`min_N` The minimum sample size per neighborhood for indices to be calculated. NA is returned for neighborhoods < `min_N`.

`max_N` Optional. The maximum sample size per neighborhood for indices to be calculated. If the number of individuals in the neighborhood exceeds `max_N`, a sample of size `max_N` will be used from the neighborhood to compute the metrics and output files specified by the user. Note that if `max_N` is specified, and the value is too small to be representative of the neighobrhood, the results could differ significantly compared to if all individuals in the neighborhood were used.

`metrics` Optional. Provide a vector of the metrics you would like sGD to produce. Options include "GD" (genetic diversity indices), "NS" (Wright's genetic neighobrhood size), "HWE" (tests for Hardy-Weinberg equilibrium, heterozygote excess, and homozygote excess), and "pFST" (a matrix of pairwise FST values for all neighborhoods)". Note that calculating pFST takes considerable time (several hours using the sGD demo data).

`NHmat_ans` Logical (Default = FALSE). If TRUE, a matrix defining neighborhood membership is written to the working directory. For each row in the matrix, a value of 1 occurs at the indices of all individuals inside the neighborhood and a value of 0 occurs for all individuals outside the neighborhood.

`genout_ans` Logical (Default = FALSE). If TRUE, a genepop file containing the genotypes for all neighborhoods is written to the working directory.

`file_name` (optional) A character string that will be appended to the front of the output filename (will end with "_sGD.csv"). If none specified, no output file will be written.

`NeEstimator_dir`
Optional. Path to the NeEstimator directory. NeEstimator 2.01 is required only if you include the "NS" metric. It can be downloaded from http://molecularfisherieslaboratory.com.au/neestimator-software.

Value

sGD returns a data frame containing estimates of genetic diversity and/or neighborhood size for neighborhoods surrounding each sample location. The order of the rows in the output matches the order of the samples in the inputs.

Variables in the output data frame include (depending on the metrics selected):

`NH_Index` - an index of the neighborhoods, from 1 to the total number of neighborhoods.

`NH_ID` - the ID of the individual at the neighborhood center, taken from individual's ID in the `xy_file`.

`X` - the X coordinate of the neighborhood center.

`Y` - the Y coordinate of the neighborhood center.

`N` - the number of individuals within the neighborhood.

`A` - the average number of alleles across all loci/individuals within the neighborhood.

`Ap` - the proportion of alleles from the entire population that area actually present in the neighborhood.

`Ar` - the allelic richness across all loci/individuals within the neighborhood.

He - the average expected heterozygosity across all loci/individuals within the neighborhood.

Ho - the average observed heterozygosity across all loci/individuals within the neighborhood.

FIS - the average inbreeding coefficient across all loci/individuals within the neighborhood.

NS_ex0 - an estimate of the effective number of breeding indviduals (Wright's neighborhood size) present within the neighborhood, not exluding rare alleles that could bias the estimate.

NS_ex0.02 - an estimate of the effective number of breeding indviduals (Wright's neighborhood size) present within the neighborhood, exluding alleles with a frequency of 0.02 or less.

NS_ex0.05 - an estimate of the effective number of breeding indviduals (Wright's neighborhood size) present within the neighborhood, exluding alleles with a frequency of 0.05 or less.

NS_ex0.10 - an estimate of the effective number of breeding indviduals (Wright's neighborhood size) present within the neighborhood, exluding alleles with a frequency of 0.10 or less.

If specified in the sGD arguments, the following output files will also be written to the working directory:

NHmat - if NHmat_and = TRUE, sGD writes the NH membership matrix described above to a .csv file in the working directory. The row and column names match the individual ID's in the input files, and are in the same order as the input files.

genout - if genout_ans = TRUE, sGD writes the NH genepop file to the working directory.

Examples

```
library(sGD)
library(adegenet)
library(raster)

# read in genotypes, locations, and distance matrix
genepop.file <- system.file("extdata","sGD_demo_IBR.gen",package="sGD")
xy = read.csv(system.file("extdata","sGD_demo_xy.csv",package="sGD"))
dist.mat <- as.matrix(read.csv(system.file("extdata","sGD_demo_cdmat.csv",package="sGD"),
                               header=FALSE))

# convert genepop to genind (make sure you specify the correct allele code digits - ncode)
genind_obj <- read.genepop(genepop.file,ncode=3L,quiet=TRUE)
pop(genind_obj) = xy$Indiv_ID # give each location a unique population ID

# run sGD
sGD_output <- sGD(genind_obj,xy,dist.mat,NH_radius=16000,min_N=20,max_N=NULL,
                  metrics=c("GD","NS","HWE"), NHmat_ans=TRUE,genout_ans=TRUE,
                  file_name="sGD_demo", NeEstimator_dir="C:/NeEstimator_2.01")

# read in the landscape raster to use in plots
landscape <- raster(system.file("extdata","sGD_demo_IBR_landscape.asc",package="sGD"))

# Convert raster to dataframe for ggplot
landscape.p <- rasterToPoints(landscape)
landscape.df <- data.frame(landscape.p)
colnames(landscape.df) <- c("X", "Y", "Resistance")

# Plot sGD output (Ap is shown here, but explore all sGD outputs) atop the resistance model
library(ggplot2)

ggplot()  +
 geom_raster(data=landscape.df,aes(x=X,y=Y,fill=Resistance),alpha=I(0.5)) +
 scale_fill_gradient(low="black", high="lightgrey") +
```

```
geom_point(data=sGD_output, aes(x=X, y=Y,color=Ap),size=5) +
scale_color_gradient(low="red", high="green",na.value = "white") +
theme(panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      panel.background = element_blank())
```

Index