Package ‘mmod’

April 6, 2017

Version 1.3.3
Date 2017-06-04
Title Modern Measures of Population Differentiation
Maintainer David Winter <david.winter@gmail.com>
Depends R (>= 2.6.0), adegenet (>= 2.0),
Imports stats, pegas
Suggests testthat
ZipData no
Description Provides functions for measuring population divergence from genotypic data.
License MIT + file LICENSE
URL https://github.com/dwinter/mmod
BugReports https://github.com/dwinter/mmod/issues
RoxygenNote 6.0.1
NeedsCompilation no
Author David Winter [aut, cre],
   Peter Green [ctb],
   Zhian Kamvar [ctb],
   Thierry Gosselin [ctb]
Repository CRAN
Date/Publication 2017-04-06 06:54:15 UTC

R topics documented:

as.genind.DNAbin ........................................... 2
chao_bootstrap ........................................... 3
diff_stats ................................................. 4
diff_test .................................................. 5
dist.codom ................................................ 6
D_Jost ....................................................... 7
as.genind.DNAbin

Description

Convert a DNAbin object into a genind object

Usage

as.genind.DNAbin(x, pops)

Arguments

x

object of class DNAbin

pops

vector of population assignments for each sequence

Value

genind

Examples

library(pegas)
data(woodmouse)
wm <- as.genind.DNAbin(woodmouse, rep(c("A", "B", "C"), each=5))
diff_stats(wm)
chao_bootstrap

Produce bootstrap samples from each subpopulation of a genind object

Description

This function produces bootstrap samples from a genind object, with each subpopulation resampled according to its size. Because there are many statistics that you may wish to calculate from these samples, this function returns a list of genind objects representing bootstrap samples that can then be further processed (see examples).

Usage

chao_bootstrap(x, nreps = 1000)

Arguments

- `x` genind object (from package adegenet)
- `nreps` numeric number of bootstrap replicates to perform (default 1000)

Details

You should note, this is a standard (frequentist) approach to quantifying uncertainty - effectively asking "if the population was exactly like our sample, and we repeatedly took samples like this from it, how much would those samples vary?" The confidence intervals don't include uncertainty produced from any biases in the way you collected your data. Additionally, this bootstrapping procedure displays a slight upward bias for some datasets. If you plan or reporting a confidence interval for your statistic, it is probably a good idea to subtract the difference between the point estimate of the statistic and the mean of the bootstrap distribution from the extremes of the interval (as demonstrated in the example below)

Value

A list of genind objects

References


See Also

Other resample: `jacknife_populations, summarise_bootstrap`
Examples

```r
## Not run:
data(nancycats)
obs.D <- D.Jost(nancycats)
bs <- chao_bootstrap(nancycats)
bs.D <- summarise_bootstrap(bs, D.Jost)
bs.D$summary.global.het - bias

## End(Not run)
```

**diff_stats**

*Calculate differentiation statistics for a genind object*

**Description**

By default this function calculates three different statistics of differentiation for a genetic dataset. Nei’s Gst, Hedrick’s G”st and Jost’s D. Optionally, it can also calculate Phi’s t, which is not calculated by default as it can take somewhat more time to run.

**Usage**

```r
diff_stats(x, phi_st = FALSE)
```

**Arguments**

- **x** genind object (from package adegenet)
- **phi_st** Boolean Calculate Phi’s t (default is FALSE)

**Details**

See individual functions (listed below) for more details.

**Value**

- per.locus values for each statistic for each locus in the dataset
- global estimates for these statistics across all loci in the dataset

**References**

diff_test


Meirmans, PW. (2005), Using the AMOVA framework to estimate a standardized genetic differentiation measure. Evolution 60: 2399-402.


See Also

Other diffstat: D_Jost, Gst_Hedrick, Gst_Nei, Phi_st_Meirmans

Examples

data(nancycats)
diff_stats(nancycats)

diff_test An exact test of population differentiation for genind objects

Description

This function uses Fisher’s exact test to determine if alleles in sub-populations are drawn randomly from a larger population (i.e. a significance test for allelic differentiation among sub-populations).

Usage

diff_test(x, sim = TRUE, nreps = 2000)

Arguments

x a genind object (from package adegenet)
sim boolean: if TRUE simulate p-value by using an MCMC sample of those tables that have the same marginal totals as the observed data (required for all but the smallest datasets)
nreps number of steps used to simulate p-value (default 2000)

Details

Note, this test returns p-values for each locus in a dataset _not_ estimates of effect size. Since most populations have some degree of population differentiation, very large samples are almost guaranteed to return significant results. Refer to estimates of the various differentiation statistics (D, G’S’T and Phi’S’T) to ascertain how meaningful such results might be.

Value

named vector of p-values testing the null hypothesis these samples were drawn from a panmictic population.
See Also

`fisher.test`, which this function wraps

Examples

```r
data(nancycats)
diff_test(seploc(nancycats)[2], nreps=100)
```

---

**dist.codom**

*Calculate distance between individual for co-dominant locus*

**Description**

This function calculates the distance between individuals in a `genind` object based on their genotypes. Specifically, the simple metric of Kosman and Leonard (2005) in which distance is calculated as a proportion of shared alleles at each locus.

**Usage**

```r
dist.codom(x, matrix = TRUE, global = TRUE, na.rm = TRUE)
```

**Arguments**

- `x`: `genind` object (from package adegenet)
- `matrix`: boolean: if `TRUE` return matrix (dist object if `FALSE`)
- `global`: boolean: if `TRUE`, return a single global estimate based on all loci. If `FALSE` return a list of matrices for each locus. if `FALSE`
- `na.rm`: boolean: if `TRUE` remove individuals with NAs

**Value**

either a list of distance matrices, one for each locus or a single matrix containing the mean distance between individuals across all loci

Dropped for each distance matrix and object of class "na.action" containing indices to those indi-

**References**

**D_Jost**

**Examples**

```r
data(nancycats)
dm <- dist.cdom(nancycats[40:45], matrix=FALSE)
head(dm)
```

**Description**

This function calculates Jost’s D from a genind object

**Usage**

```r
D_Jost(x, hsht_mean = "arithmetic")
```

**Arguments**

- `x`: genind object (from package adegenet)
- `hsht_mean`: The type of mean to use to calculate values of Hs and Ht for a global estimate. (Default is arithmetic mean, can also be set to the harmonic mean).

**Details**

Takes a genind object with population information and calculates Jost’s D Returns a list with values for each locus as well as two global estimates. ’global.het’ uses the averages of Hs and Ht across all loci while ’global.harm_mean’ takes the harmonic mean of all loci.

Because estimators of Hs and Ht are used, its possible to have negative estimates of D. You should treat these as numbers close to zero.

**Value**

- per.locus values for each D for each locus in the dataset
- global estimtes for D based on overall heterozygosity or the harmonic mean of values for each locus

**References**


**See Also**

Other diffstat: Gst_Hedrick, Gst_Nei, Phi_st_Meirmans, diff_stats

Other D: pairwise_D
Examples

data(nancycats)
D_Jost(nancycats)
D_Jost(nancycats, hsht_mean= "arithmetic")

Gst_Hedrick

Calculate Nei's Gst using estimators for Hs and Ht

Description
This function calculates Hedrick’s G’st from a genind object

Usage
Gst_Hedrick(x)

Arguments
x
  genind object (from package adegenet)

Details
Takes a genind object with population information and calculates Hedrick’s G”st. Because estimators of Hs and Ht are used, it’s possible to have negative estimates of G”st. You should treat such results as zeros (or an attempt to estimate a very low number with some error which might push it below zero)

Value
per.locus values for each G”st for each locus in the dataset
global estinates for G”st based on overall heterozygosity

References

See Also
Other diffstat: D_Jost, Gst_Nei, Phi_st_Meirmans, diff_stats
Other Hedrick: pairwise_Gst_Hedrick

Examples

data(nancycats)
Gst_Hedrick(nancycats)
Gst_Nei

Calculate Nei’s Gst using estimators for Hs and Ht

Description

This function calculates Gst following Nei’s method and using Nei and Chesser’s estimators for Hs and Ht

Usage

Gst_Nei(x)

Arguments

x genind object (from package adegenet)

Value

per.locus estimates of Gst for each locus in the dataset
per.locus estimates of Gst for across all loci

References


See Also

Other diffstat: D_Jost,Gst_Hedrick,Phi_st_Meirmans,diff_stats

Other Nei: pairwise_Gst_Nei

Examples

data(nancycats)
Gst_Nei(nancycats)
### harmonic_mean

**Harmonic mean**

**Description**

Calculate the harmonic mean of a numeric vector (will return NA if there are any negative numbers in the vector).

**Usage**

```r
harmonic_mean(x, na.rm = TRUE)
```

**Arguments**

- `x`: numeric vector
- `na.rm`: logical, remove NAs prior or calculation

**Value**

Harmonic mean of vector

**Examples**

```r
data(nancycats)
pop_sizes <- table(pop(nancycats))
harmonic_mean(pop_sizes)
```

### jacknife_populations

**Create jacknife samples of a genind object by population**

**Description**

Makes a series of jacknife samples across populations from a genind object. This function returns a list of genind objects that can then be further processed (see examples below).

**Usage**

```r
jacknife_populations(x, sample_frac = 0.5, nreps = 1000)
```

**Arguments**

- `x`: genind object (from package adegenet)
- `sample_frac`: fraction of pops to sample in each replication (default 0.5)
- `nreps`: number of jacknife replicates to run (default 1000)
mmod

Value

a list of genind objects to be further processed

See Also

Other resample: chao_bootstrap, summarise_bootstrap

Examples

```r
## Not run:
data(nancycats)
obsv <- diff_stats(nancycats)
jn <- jackknife_populations(nancycats)
jn.D <- summarise_bootstrap(jn, D_Jost)
## End(Not run)
```

mmod

Modern Measures of Differentiation

Description

Population geneticists have traditionally used Nei’s Gst (often confusingly called Fst...) to measure divergence between populations. Recently, it has become clear that simple interpretations of the value of Gst can be misleading. For this reason several new measures of differentiation have been developed. mmod is a package that brings some of these measures to R.

Details

The vignette for this package (available using vignette("demo", package="mmod") from within R) contains an introduction to these methods and example usage for this package. I strongly suggest new users start by reading this documentation.

pairwise_D

Calculates pairwise values of Jost’s D

Description

This function calculates Jost’s D, a measure of genetic differentiation, between all combinations of populations in a genind object.

Usage

`pairwise_D(x, linearized = FALSE, hsht_mean = "arithmetic")`
Arguments

- `x` - genind object (from package adegenet)
- `linearized` - logical, if TRUE will turned linearized D (1/(1-D))
- `hshft_mean` - type of mean to use for the global estimates of Hs and Ht default it "arithmetic", can also be set to "harmonic".

Value

A distance matrix with between-population values of D

References


See Also

Other pairwise: `pairwise_Gst_Hedrick`, `pairwise_Gst_Nei`
Other D: `D_Jost`

Examples

```r
data(nancycats)
pairwise_D(nancycats[1:26,])
```

**pairwise_Gst_Hedrick**  
Calculates pairwise values of Hedrick's G'st

Description

This function calculates Hedrick's G'st, a measure of genetic differentiation, between all combinations of populations in a genind object.

Usage

`pairwise_Gst_Hedrick(x, linearized = FALSE)`

Arguments

- `x` - genind object (from package adegenet)
- `linearized` - logical, if TRUE will turned linearized G'st (1/(1-G'st))

Value

A distance matrix with between-population values of G"st
pairwise_Gst_Nei

References

See Also
Other pairwise: pairwise_D, pairwise_Gst_Nei
Other Hedrick: Gst_Hedrick

Examples

```r
data(nancycats)
pairwise_Gst_Hedrick(nancycats[1:26,])
```

---

**pairwise_Gst_Nei**

*Calculates pairwise values of Nei's Gst*

Description
This function calculates Nei's Gst, a measure of genetic differentiation, between all combinations of populations in a genind object.

Usage

```r
pairwise_Gst_Nei(x, linearized = FALSE)
```

Arguments

- **x**: genind object (from package adegenet)
- **linearized**: logical, if TRUE will turn linearized Gst (1/(1-Gst))

Value

dist A distance matrix with between-population values of Gst

References

See Also

Other pairwise: pairwise_D, pairwise_Gst_Hedrick
Other Nei: Gst_Nei
**Description**

This function calculates Meirmans' corrected version of Phi_st, an Fst analog produced using the AMOVA framework. Note, the global estimate produced by this function is calculated as the mean distance between individuals across all loci, and this excluded individuals with one or more missing value.

**Usage**

`Phi_st_Meirmans(x)`

**Arguments**

- `x` genind object (from package adegenet)

**Value**

- per.locus Phi_st estimate for each locus
- global Phi_st estimate across all loci

**References**

Meirmans, PW. (2005), Using the AMOVA framework to estimate a standardized genetic differentiation measure. Evolution 60: 2399-402.


**See Also**

Other diffstat: `D_Jost,Gst_Hedrick,Gst_Nei,diff_stats`

**Examples**

```r
data(nancycats)
Phi_st_Meirmans(nancycats[1:26,])
```
rgenotypes  

Randomly create genotypes

Description

Use the multinomial distribution to randomly create genotypes for individuals for given allele frequencies. By default this function returns a matrix of with alleles in rows and individuals in columns. There is an option to return a genind object representing the same data (see examples).

Usage

rgenotypes(n, ploidy, probs, genind = FALSE, pop_name = "A", loc_name = "L1")

Arguments

- n  integer number of individuals.
- ploidy  integer number of alleles to assign to each individual.
- probs  vector of probabilities corresponding to allele frequencies.
- genind  boolean if TRUE return a genind object
- pop_name  character Name for population defined in genind object (not required if genind is not TRUE)
- loc_name  character name to five locus in genind object

Details

Used in chao_bootstrap, also exported as it may come in handy for other simulations.

Value

Either a matrix with individuals in columns, alleles in rows or, if genind is TRUE a genind object for one population and locus.

See Also

rmultinom which this function wraps.

Examples

data(nancycats)
obs_allele_freqs <- apply(nancycats$tab[,1:16], 2,mean, na.rm=TRUE)
rgenotypes(10, 2, obs_allele_freqs)
summarise_bootstrap

Apply a differentiation statistic to a bootstrap sample

Description
This function applies a differentiation statistic (e.g., D_Jost, Gst_Hedrick or Gst_Nei) to a list of genind objects, possibly produced with chao_bootstrap or jackknife_populations.

Usage
summarise_bootstrap(bs, statistic)

Arguments
bs list of genind objects
statistic differentiation statistic to apply (the function itself, as with apply family functions)

Details
Two different approaches are used for calculating confidence intervals in the results. The estimates given by lowerNpercentile and upperNpercentile are simply the 2.5th and 97.5th percentile of the statistic across bootstrap samples. Note, the presence or rare alleles in some populations can bias bootstrapping procedures such that these intervals are not centered on the observed value. The mean of statistic across samples is returned as meanNbs and can be used to correct biased bootstrap samples. Alternatively, lowerNnormal and upperNnormal form a confidence interval centered on the observed value of the statistic and using the standard deviation of the statistic across replicates to generate limits (sometimes called the normal-method of obtaining a confidence interval). The print function for objects returned by this function displays the normal-method confidence intervals.

Value
per.locus: matrix of statistics calculated for each locus (column) and each bootstrap replicate (row).
global.het: vector of global estimates calculated from overall heterozygosity
global.het: vector of global estimates calculated from harmonic mean of statistic (only applied to D_Jost)
summary.loci: data.frame summarising the distribution of the chosen statistic across replicates. Details of the different confidence intervals are given in details
summary.global_het: A vector containing the same measures as summary.loci but for a global value of the statistic calculated from all loci
summary.global_harm: As with summary.global_het but calculated from the harmonic mean of the statistic across loci (only applies to D_Jost)

See Also
Other resample: chao_bootstrap, jackknife_populations
**Examples**

```r
## Not run:
data(nancycats)
bs <- chao_bootstrap(nancycats)
summarise_bootstrap(bs, D_Jost)

## End(Not run)
```
Index

as.genind.DNAbin, 2

chao_bootstrap, 3, 11, 15, 16

D_Jost, 5, 7, 8, 9, 12, 14
diff_stats, 4, 7–9, 14
diff_test, 5
dist.codom, 6

fisher.test, 6

Gst_Hedrick, 5, 7, 8, 9, 13, 14
Gst_Nei, 5, 7, 8, 9, 13, 14

harmonic_mean, 10

jackknife_populations, 3, 10, 16

mmod, 11
mmod-package (mmod), 11

pairwise_D, 7, 11, 13
pairwise_Gst_Hedrick, 8, 12, 12, 13
pairwise_Gst_Nei, 9, 12, 13, 13
Phi_st_Meirmans, 5, 7–9, 14

rgenotypes, 15
rmultinom, 15

summarise_bootstrap, 3, 11, 16