Package ‘mmod’

July 23, 2025

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

Contents

`as.genind.DNAbin` *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 assignemnts 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 calculte from these samples, this function returns a list of genind objects representing bootsrap samples that can then be futher 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 boostrapping 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 boostrap distribution from the extremes of the interval (as demonstrated in the expample below)

Value

A list of genind objects

References

Chao, A. et al. (2008). A Two-Stage probabilistic approach to Multiple-Community similarity indices. Biometrics, 64:1178-1186

See Also

Other resample: `jacknife_populations`, `summarise_bootstrap`

Examples

```
## Not run:
data(nancycats)
obs.D <- D_Jost(nancycats)
bs <- chao_bootstrap(nancycats)
bs_D <- summarise_bootstrap(bs, D_Jost)
bias <- bs.D$summary.global.het[1] - obs.D$global.het
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'st, which is not calculated by default as it can take somewhat more time to run.

Usage

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

Arguments

`x`	genind object (from package adegenet)
`phi_st`	Boolean Calculate Phi_st (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 estimtes for these statistics across all loci in the dataset

References

Hedrick, PW. (2005), A Standardized Genetic Differentiation Measure. Evolution 59: 1633-1638.

Jost, L. (2008), GST and its relatives do not measure differentiation. Molecular Ecology, 17: 4015-4026.

Meirmans PG, Hedrick PW (2011), Assessing population structure: FST and related measures. Molecular Ecology Resources, 11:5-18

Nei M. (1973) Analysis of gene diversity in subdivided populations. PNAS: 3321-3323.

Nei M, Chesser RK. (1983). Estimation of fixation indices and gene diversities. Annals of Human Genetics. 47: 253-259.

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

Excoffier, L., Smouse, P., Quattro, J. (1992), Analysis of molecular variance inferred from metric distances among DNA haplotypes: application to human mitochondrial DNA restriction data. Genetics 131: 479-91

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”ST and Phi’ST)to ascertain how meaningful such results might be.

Value

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

See Also

fisher.test, which this function wraps

Examples

```
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 propotion of shared alleles at each locus.

Usage

```
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 indivudals in the genind object which where omitted due to having NAs

References

Kosman E., Leonard, K.J. Similarity coefficients for molecular markers in studies of genetic relationships between individuals for haploid diploid, and polyploid species. Molecular Ecology. 14: 415-424

Examples

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

`D_Jost` *Calculate Jost's D*

Description

This function calculates Jost's D from a genind object

Usage

```
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 teh airthmetic 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

Jost, L. (2008), GST and its relatives do not measure differentiation. Molecular Ecology, 17: 4015-4026.

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 estimtes for G"st based on overall heterozygosity

References

Hedrick, PW. (2005), A Standardized Genetic Differentiation Measure. Evolution 59: 1633-1638.

Meirmans PG, Hedrick PW (2011), Assessing population structure: FST and related measures. Molecular Ecology Resources, 11:5-18

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

Nei M. (1973) Analysis of gene diversity in subdivided populations. PNAS: 3321-3323.

Nei M, Chesser RK. (1983). Estimation of fixation indices and gene diversities. Annals of Human Genetics. 47: 253-259.

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

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

Arguments

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

Value

harmonic mean of vector

Examples

```
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

```
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)

Value

a list of genind objects to be further processed

See Also

Other resample: `chao_bootstrap`, `summarise_bootstrap`

Examples

```
## Not run:
data(nancycats)
obs <- diff_stats(nancycats)
jn <- jacknife_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 intereptations of the value of Gst can be misleading. For this reason several new measures differntiation have been developed. mmod is a package that brings some of these measures to R.

Details

The vignette for this package (avaliable using vignette("demo", package="mmod") from within R) contains an introduction to these methods and 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 populaitons 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)
`hsht_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

Jost, L. (2008), GST and its relatives do not measure differentiation. Molecular Ecology, 17: 4015-4026.

See Also

Other pairwise: `pairwise_Gst_Hedrick`, `pairwise_Gst_Nei`

Other D: `D_Jost`

Examples

```
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 populaitons 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

References

Hedrick, PW. (2005), A Standardized Genetic Differentiation Measure. Evolution 59: 1633-1638.

See Also

Other pairwise: `pairwise_D`, `pairwise_Gst_Nei`

Other Hedrick: `Gst_Hedrick`

Examples

```
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 populaitons in a genind object.

Usage

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

Arguments

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

Value

dist A distance matrix with between-population values of Gst

References

Nei M. (1973) Analysis of gene diversity in subdivided populations. PNAS: 3321-3323.

Nei M, Chesser RK. (1983). Estimation of fixation indices and gene diversities. Annals of Human Genetics. 47: 253-259.

See Also

Other pairwise: `pairwise_D`, `pairwise_Gst_Hedrick`

Other Nei: `Gst_Nei`

Examples

```
data(nancycats)
pairwise_Gst_Nei(nancycats[1:26,])
```

`Phi_st_Meirmans` *Calculate Phi_st from a genind object*

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 exlcuded 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.

Excoffier, L., Smouse, P., Quattro, J. (1992), Analysis of molecular variance inferred from metric distances among DNA haplotypes: application to human mitochondrial DNA restriction data. Genetics 131: 479-91

See Also

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

Examples

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

`rgenotypes` *Randomly create genotypes*

Description

Use the multinomial distribution to randomly create genotpes for individuals for given allele frequences. 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 indviduals.
`ploidy`	integer number of alleles to asign to each individual.
`probs`	vector of probabilies corresponding to allele frequences.
`genind`	boolean if TRUE return a genind object
`pop_name`	charcter 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 (eg, D_Jost, Gst_Hedrick or Gst_Nei) to a list of genind objects, possibly produced with chao_bootsrap or jacknife_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 `lower.percentile` and `upper.percentile` are simply the `2.5`th and `97.5`th precentile 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 `mean.bs` and can be used to correct biased bootsrap samples. Alternatively, `lower.normal` and `upper.normal` 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: `matirx` 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`, `jacknife_populations`

Examples

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

## End(Not run)
```

Index