Package 'FinePop'

March 15, 2024

Type Package

Title Fine-Scale Population Analysis

Version 1.5.2

Date 2024-03-15

Author Reiichiro Nakamichi, Hirohisa Kishino, Shuichi Kitada

Maintainer Reiichiro Nakamichi <nakamichi_reiichiro33@fra.go.jp>

Description Statistical tool set for population genetics. The package provides following functions: 1) empirical Bayes estimator of Fst and other measures of genetic differentiation, 2) regression analysis of environmental effects on genetic differentiation using bootstrap method, 3) interfaces to read and manipulate 'GENEPOP' format data files and allele/haplotype frequency format files.

License GPL (>= 2.0)

Depends R (>= 4.0.0)

Suggests ape

LazyLoad yes

NeedsCompilation no

Encoding UTF-8

Repository CRAN

Date/Publication 2024-03-15 10:40:06 UTC

R topics documented:

FinePop-package   *Fine-Scale Population Analysis*

Description

Statistical tool set for population genetics. The package provides following functions: 1) empirical Bayes estimator of Fst and other measures of genetic differentiation, 2) regression analysis of environmental effects on genetic differentiation using bootstrap method, 3) interfaces to read and manipulate 'GENEPOP' format data files and allele/haplotype frequency format files.

Author(s)

Reiichiro Nakamichi, Hirohisa Kishino, Shuichi Kitada
Maintainer: Reiichiro Nakamichi <nakamichi_reiichiro33@fra.go.jp>

References

Kitada S, Kitakado T, Kishino H (2007) Empirical Bayes inference of pairwise FST and its distribution in the genome. *Genetics*, 177, 861-873.

Kitada S, Nakamichi R, Kishino H (2017) The empirical Bayes estimators of fine-scale population structure in high gene flow species. *Mol. Ecol. Resources*, DOI: 10.1111/1755-0998.12663

Rousset F (2008) Genepop'007: a complete reimplementation of the Genepop software for Windows and Linux. *Mol. Ecol. Resources*, 8, 103-106.

clip.genepop   *Remove designated markers from a GENEPOP file.*

Description

This function reads a GENEPOP file (Rousset 2008), remove designated markers, and write a GENEPOP file of clipped data. The user can directly designate the names of the markers to be removed. The user also can set the filtering threshold of major allele frequency.

Usage

```
clip.genepop(infile, outfile, remove.list = NULL, major.af = NULL)
```

Arguments

infile	A character value specifying the name of the GENEPOP file to be clipped.
outfile	A character value specifying the name of the clipped GENEPOP file.
remove.list	A character value or vector specifying the names of the markers to be removed. The names must be included in the target GENEPOP file.
major.af	A numeric value specifying the threshold of major allele frequency for marker removal. Markers with major allele frequencies higher than this value will be removed. This value must be between 0 and 1.

Author(s)

Reiichiro Nakamichi

References

Rousset F (2008) Genepop'007: a complete reimplementation of the Genepop software for Windows and Linux. *Mol. Ecol. Resources*, 8, 103-106.

Examples

```
# Example of GENEPOP file
data(jsmackerel)
jsm.genepop.file <- tempfile()
cat(jsmackerel$MS.genepop, file=jsm.genepop.file, sep="\n")

# Remove markers designated by their names
clipped_by_name.jsm.genepop.file <- tempfile()
clip.genepop(infile=jsm.genepop.file,
             outfile=clipped_by_name.jsm.genepop.file,
             remove.list=c("Sni21","Sni26"))

# Remove markers with high major allele frequencies (in this example, > 0.5)
clipped_by_af.jsm.genepop.file <- tempfile()
clip.genepop(infile=jsm.genepop.file,
             outfile=clipped_by_af.jsm.genepop.file,
             major.af=0.5)

# Remove markers both by their names and by major allele frequencies
clipped_by_both.jsm.genepop.file <- tempfile()
clip.genepop(infile=jsm.genepop.file,
             outfile=clipped_by_both.jsm.genepop.file,
             remove.list=c("Sni21","Sni26"), major.af=0.5)

# See four text files in temporary directory.
#  jsm.genepop.file                 : original data of five markers
#  clipped_by_name.jsm.genepop.file : clipped data by marker names
```

```
# clipped_by_af.jsm.genepop.file   : clipped data by allele frequency
# clipped_by_both.jsm.genepop.file : clipped data by both names and frequency
```

DJ *Jost's D*

Description

This function estimates pairwise D (Jost 2008) among subpopulations from a GENEPOP data object (Rousset 2008). Missing genotype values in the GENEPOP file ("0000" or "000000") are simply ignored.

Usage

```
DJ(popdata)
```

Arguments

popdata   Population data object created by read.genepop function from a GENEPOP file.

Value

Matrix of estimated pairwise Jost's D.

Author(s)

Reiichiro Nakamichi, Hirohisa Kishino, Shuichi Kitada

References

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

Rousset F (2008) Genepop'007: a complete reimplementation of the Genepop software for Windows and Linux. *Mol. Ecol. Resources*, 8, 103-106.

See Also

read.genepop

Examples

```
# Example of GENEPOP file
data(jsmackerel)
jsm.ms.genepop.file <- tempfile()
jsm.popname.file <- tempfile()
cat(jsmackerel$MS.genepop, file=jsm.ms.genepop.file, sep="\n")
cat(jsmackerel$popname, file=jsm.popname.file, sep=" ")
```

```
# Data load
# Prepare your GENEPOP file and population name file in the working directory
# Replace "jsm.ms.genepop.file" and "jsm.popname.file" by your file names.
popdata <- read.genepop(genepop=jsm.ms.genepop.file, popname=jsm.popname.file)

# Jost's D estimation
result.DJ <- DJ(popdata)
print(as.dist(result.DJ))
```

EBDJ   *Empirical Bayes estimator of Jost's D*

Description

This function estimates pairwise D (Jost 2008) among subpopulations using empirical Bayes method (Kitada et al. 2007). This function accepts two types of data object, GENEPOP data (Rousset 2008) and allele (haplotype) frequency data (Kitada et al. 2007). Missing genotype values in the GENEPOP file ("0000" or "000000") are simply ignored.

Usage

```
EBDJ(popdata, num.iter=100)
```

Arguments

popdata   Genotype data object of populations created by read.genepop function from a GENEPOP file. Allele (haplotype) frequency data object created by read.frequency function from a frequency format file also is acceptable.

num.iter   A positive integer value specifying the number of iterations in empirical Bayes simulation.

Details

Frequency format file is a plain text file containing allele (haplotype) count data. This format is mainly for mitochondrial DNA (mtDNA) haplotype frequency data, however nuclear DNA (nDNA) data also is applicable. In the data object created by read.frequency function, "number of samples" means haplotype count. Therefore, it equals the number of individuals in mtDNA data, however it is the twice of the number of individuals in nDNA data. First part of the frequency format file is the number of subpopulations, second part is the number of loci, and latter parts are [*population* x *allele*] matrices of the observed allele (haplotype) counts at each locus. Two examples of frequency format files are attached in this package. See `jsmackerel`.

Value

Matrix of estimated pairwise Jost's D.

Author(s)

Reiichiro Nakamichi, Hirohisa Kishino, Shuichi Kitada

References

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

Kitada S, Kitakado T, Kishino H (2007) Empirical Bayes inference of pairwise FST and its distribution in the genome. *Genetics*, 177, 861-873.

Rousset F (2008) Genepop'007: a complete reimplementation of the Genepop software for Windows and Linux. *Mol. Ecol. Resources*, 8, 103-106.

See Also

read.genepop, read.frequency

Examples

```
# Example of GENEPOP file
data(jsmackerel)
jsm.ms.genepop.file <- tempfile()
jsm.popname.file <- tempfile()
cat(jsmackerel$MS.genepop, file=jsm.ms.genepop.file, sep="\n")
cat(jsmackerel$popname, file=jsm.popname.file, sep=" ")

# Data load
# Prepare your GENEPOP file and population name file in the working directory
# Replace "jsm.ms.genepop.file" and "jsm.popname.file" by your file names.
popdata <- read.genepop(genepop=jsm.ms.genepop.file, popname=jsm.popname.file)

# Jost's D estimation
result.EBDJ <- EBDJ(popdata)
print(as.dist(result.EBDJ))
```

EBFST   *Empirical Bayes estimator of Fst.*

Description

This function estimates global/pairwise Fst among subpopulations using empirical Bayes method (Kitada et al. 2007, 2017). Preciseness of estimated pairwise Fst is evaluated by bootstrap method. This function accepts two types of data object, GENEPOP data (Rousset 2008) and allele (haplotype) frequency data (Kitada et al. 2007). Missing genotype values in the GENEPOP file ("0000" or "000000") are simply ignored.

Usage

```
EBFST(popdata, num.iter = 100, locus = F)
```

Arguments

popdata	Genotype data object of populations created by read.genepop function from a GENEPOP file. Allele (haplotype) frequency data object created by read.frequency function from a frequency format file also is acceptable.
num.iter	A positive integer value specifying the number of iterations in empirical Bayes simulation.
locus	A Logical argument indicating whether locus-specific Fst values should be calculated.

Details

Frequency format file is a plain text file containing allele (haplotype) count data. This format is mainly for mitochondrial DNA (mtDNA) haplotype frequency data, however nuclear DNA (nDNA) data also is applicable. In the data object created by read.frequency function, "number of samples" means haplotype count. Therefore, it equals the number of individuals in mtDNA data, however it is the twice of the number of individuals in nDNA data. First part of the frequency format file is the number of subpopulations, second part is the number of loci, and latter parts are [*population* x *allele*] matrices of the observed allele (haplotype) counts at each locus. Two examples of frequency format files are attached in this package. See `jsmackerel`.

Value

global:

theta	Estimated gene flow rate.
fst	Estimated genome-wide global Fst.
fst.locus	Estimated locus-specific global Fst. (If locus = TRUE)

pairwise:

fst	Estimated genome-wide pairwise Fst.
fst.boot	Bootstrap mean of estimated Fst.
fst.boot.sd	Bootstrap standard deviation of estimated Fst.
fst.locus	Estimated locus-specific pairwise Fst. (If locus = TRUE)

Author(s)

Reiichiro Nakamichi, Hirohisa Kishino, Shuichi Kitada

References

Kitada S, Kitakado T, Kishino H (2007) Empirical Bayes inference of pairwise FST and its distribution in the genome. *Genetics*, 177, 861-873.

Kitada S, Nakamichi R, Kishino H (2017) The empirical Bayes estimators of fine-scale population structure in high gene flow species. *Mol. Ecol. Resources*, DOI: 10.1111/1755-0998.12663

Rousset F (2008) Genepop'007: a complete reimplementation of the Genepop software for Windows and Linux. *Mol. Ecol. Resources*, 8, 103-106.

See Also

read.genepop, read.frequency, as.dist, as.dendrogram, hclust, cmdscale, nj

Examples

```
# Example of GENEPOP file
data(jsmackerel)
jsm.ms.genepop.file <- tempfile()
jsm.popname.file <- tempfile()
cat(jsmackerel$MS.genepop, file=jsm.ms.genepop.file, sep="\n")
cat(jsmackerel$popname, file=jsm.popname.file, sep=" ")

# Data load
# Prepare your GENEPOP file and population name file in the working directory
# Replace "jsm.ms.genepop.file" and "jsm.popname.file" by your file names.
popdata <- read.genepop(genepop=jsm.ms.genepop.file, popname=jsm.popname.file)

# Fst estimation
result.eb <- EBFST(popdata)
ebfst <- result.eb$pairwise$fst
ebfst.d <- as.dist(ebfst)
print(ebfst.d)

# dendrogram
ebfst.hc <- hclust(ebfst.d,method="average")
plot(as.dendrogram(ebfst.hc), xlab="",ylab="",main="", las=1)

# MDS plot
mds <- cmdscale(ebfst.d)
plot(mds, type="n", xlab="",ylab="")
text(mds[,1],mds[,2], popdata$pop_names)

# NJ tree
library(ape)
ebfst.nj <- nj(ebfst.d)
plot(ebfst.nj,type="u",main="",sub="")
```

EBGstH   *Empirical Bayes estimator of Hedrick's G'st*

Description

This function estimates pairwise G'st (Hedrick 2005) among subpopulations using empirical Bayes method (Kitada et al. 2007). This function accepts two types of data object, GENEPOP data (Rousset 2008) and allele (haplotype) frequency data (Kitada et al. 2007). Missing genotype values in the GENEPOP file ("0000" or "000000") are simply ignored.

Usage

```
EBGstH(popdata, num.iter = 100)
```

Arguments

popdata	Genotype data object of populations created by read.genepop function from a GENEPOP file. Allele (haplotype) frequency data object created by read.frequency function from a frequency format file also is acceptable.
num.iter	A positive integer value specifying the number of iterations in empirical Bayes simulation.

Details

Frequency format file is a plain text file containing allele (haplotype) count data. This format is mainly for mitochondrial DNA (mtDNA) haplotype frequency data, however nuclear DNA (nDNA) data also is applicable. In the data object created by read.frequency function, "number of samples" means haplotype count. Therefore, it equals the number of individuals in mtDNA data, however it is the twice of the number of individuals in nDNA data. First part of the frequency format file is the number of subpopulations, second part is the number of loci, and latter parts are [*population* x *allele*] matrices of the observed allele (haplotype) counts at each locus. Two examples of frequency format files are attached in this package. See `jsmackerel`.

Value

Matrix of estimated pairwise Hedrick's G'st.

Author(s)

Reiichiro Nakamichi, Hirohisa Kishino, Shuichi Kitada

References

Hedrick P (2005) A standardized genetic differentiation measure. *Evolution*, 59, 1633-1638.

Kitada S, Kitakado T, Kishino H (2007) Empirical Bayes inference of pairwise FST and its distribution in the genome. *Genetics*, 177, 861-873.

Rousset F (2008) Genepop'007: a complete reimplementation of the Genepop software for Windows and Linux. *Mol. Ecol. Resources*, 8, 103-106.

See Also

`read.genepop`, `read.frequency`

Examples

```
# Example of GENEPOP file
data(jsmackerel)
jsm.ms.genepop.file <- tempfile()
jsm.popname.file <- tempfile()
cat(jsmackerel$MS.genepop, file=jsm.ms.genepop.file, sep="\n")
cat(jsmackerel$popname, file=jsm.popname.file, sep=" ")

# Data load
```

```
# Prepare your GENEPOP file and population name file in the working directory
# Replace "jsm.ms.genepop.file" and "jsm.popname.file" by your file names.
popdata <- read.genepop(genepop=jsm.ms.genepop.file, popname=jsm.popname.file)

# Hedrick's G'st estimation
result.EBGstH <- EBGstH(popdata)
print(as.dist(result.EBGstH))
```

FstBoot   *Bootstrap sampler of Fst*

Description

This function provides bootstrapped estimators of Fst to evaluate the environmental effects on the
genetic diversity. See Details.

Usage

```
FstBoot(popdata, fst.method = "EBFST", bsrep = 100, log.bs = F, locus = F)
```

Arguments

popdata	Genotype data object of populations created by read.genepop function from a GENEPOP file.
fst.method	A character value specifying the Fst estimation method to be used. Currently, "EBFST", "EBGstH", "EBDJ", "GstN", "GstNC", "GstH", "DJ" and "thetaWC.pair" are available.
bsrep	A positive integer value specifying the trial times of bootstrapping.
log.bs	A logical value specifying whether the bootstrapped data of each trial should be saved. If TRUE, GENEPOP format files named "gtdata_bsXXX.txt" (XXX=trial number) are saved in the working directory.
locus	A Logical argument indicating whether locus-specific Fst values should be calculated.

Details

FinePop provides a method for regression analyses of the pairwise Fst values against geographical
distance and the differences to examine the effect of environmental variables on population differ-
entiation (Kitada et. al 2017). First, `FstBoot` function resamples locations with replacement, and
then, we also resample the member individuals with replacement from the sampled populations. It
calculates pairwise Fst for each bootstrap sample. Second, `FstEnv` function estimates regression
coefficients (`lm` function) of for the Fst values for each iteration. It then computes the standard
deviation of the regression coefficients, Z-values and P-values of each regression coefficient. All
possible model combinations for the environmental explanatory variables were examined, includ-
ing their interactions. The best fit model with the minimum information criterion (TIC, Takeuchi
1976, Burnham & Anderson 2002) is selected. Performance for detecting environmental effects on
population structuring is evaluated by the R2 value.

Value

`bs.pop.list`	List of subpopulations in bootstrapped data
`bs.fst.list`	List of genome-wide pairwise Fst matrices for bootstrapped data.
`org.fst`	Genome-wide pairwise Fst matrix for original data.
`bs.fst.list.locus`	
	List of locus-specific pairwise Fst matrices for bootstrapped data. (If locus = TRUE)
`org.fst.locus`	Locus-specific pairwise Fst matrix for original data. (If locus = TRUE)

Author(s)

Reiichiro Nakamichi, Hirohisa Kishino, Shuichi Kitada

References

Burnham KP, Anderson DR (2002) *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach.* Springer, New York.

Kitada S, Nakamichi R, Kishino H (2017) The empirical Bayes estimators of fine-scale population structure in high gene flow species. *Mol. Ecol. Resources*, DOI: 10.1111/1755-0998.12663.

Takeuchi K (1976) Distribution of information statistics and criteria for adequacy of Models. *Mathematical Science*, 153, 12-18 (in Japanese).

See Also

read.genepop, FstEnv, EBFST, EBGstH,EBDJ, GstN, GstNC, GstH, DJ, thetaWC.pair, herring

Examples

```
# Example of genotypic and environmental dataset
data(herring)

# Data bootstrapping and Fst estimation
# fstbs <- FstBoot(herring$popdata)

# Effects of environmental factors on genetic differentiation
# fstenv <- FstEnv(fstbs, herring$environment, herring$distance)

# Since these calculations are too heavy, pre-caluculated results are included in this dataset.
fstbs <- herring$fst.bootstrap
fstenv <- herring$fst.env
summary(fstenv)
```

FstEnv   *Regression analysis of environmental factors on genetic differentiation*

Description

This function provides linear regression analysis for Fst against environmental factors to evaluate the environmental effects on the genetic diversity. See Details.

Usage

```
FstEnv(fst.bs, environment, distance = NULL)
```

Arguments

fst.bs	Bootstrap samples of pairwise Fst matrices provided by FstBoot function from a GENEPOP file.
environment	A table object of environmental factors. Rows are subpopulations, and columns are environmental factors. Names of subpopulations (row names) must be same as those in fst.bs.
distance	A square matrix of distance among subpopulations (omittable). Names of sub-populations (row/column names) must be same as those in fst.bs.

Details

FinePop provides a method for regression analyses of the pairwise Fst values against geographical distance and the differences to examine the effect of environmental variables on population differentiation (Kitada et. al 2017). First, `FstBoot` function resamples locations with replacement, and then, we also resample the member individuals with replacement from the sampled populations. It calculates pairwise Fst for each bootstrap sample. Second, `FstEnv` function estimates regression coefficients (`lm` function) of for the Fst values for each iteration. It then computes the standard deviation of the regression coefficients, Z-values and P-values of each regression coefficient. All possible model combinations for the environmental explanatory variables were examined, including their interactions. The best fit model with the minimum Takeuchi information criterion (TIC, Takeuchi 1976, Burnham & Anderson 2002) is selected. Performance for detecting environmental effects on population structuring is evaluated by the R2 value.

Value

A list of regression result:

model	Evaluated model of environmental factors on genetic differentiation.
coefficients	Estimated coefficient, standard deviation, Z value and p value of each factor.
TIC	Takeuchi information criterion.
R2	coefficient of determination.

Author(s)

Reiichiro Nakamichi, Hirohisa Kishino, Shuichi Kitada

References

Burnham KP, Anderson DR (2002) *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach.* Springer, New York.

Kitada S, Nakamichi R, Kishino H (2017) The empirical Bayes estimators of fine-scale population structure in high gene flow species. *Mol. Ecol. Resources*, DOI: 10.1111/1755-0998.12663.

Takeuchi K (1976) Distribution of information statistics and criteria for adequacy of Models. *Mathematical Science*, 153, 12-18 (in Japanese).

See Also

FstBoot, read.genepop, lm, herring

Examples

```
# Example of genotypic and environmental dataset
data(herring)

# Data bootstrapping and Fst estimation
# fstbs <- FstBoot(herring$popdata)

# Effects of environmental factors on genetic differentiation
# fstenv <- FstEnv(fstbs, herring$environment, herring$distance)

# Since these calculations are too heavy, pre-calculated results are included in this dataset.
fstbs <- herring$fst.bootstrap
fstenv <- herring$fst.env
summary(fstenv)
```

GstH   *Hedrick's G'st*

Description

This function estimates pairwise G'st (Hedrick 2005) among subpopulations from a GENEPOP data object (Rousset 2008). Missing genotype values in the GENEPOP file ("0000" or "000000") are simply ignored.

Usage

```
GstH(popdata)
```

Arguments

popdata   Population data object created by read.genepop function from a GENEPOP file.

Value

Matrix of estimated pairwise Hedrick's G'st.

Author(s)

Reiichiro Nakamichi, Hirohisa Kishino, Shuichi Kitada

References

Hedrick P (2005) A standardized genetic differentiation measure. *Evolution*, 59, 1633-1638.

Rousset F (2008) Genepop'007: a complete reimplementation of the Genepop software for Windows and Linux. *Mol. Ecol. Resources*, 8, 103-106.

See Also

read.genepop

Examples

```
# Example of GENEPOP file
data(jsmackerel)
jsm.ms.genepop.file <- tempfile()
jsm.popname.file <- tempfile()
cat(jsmackerel$MS.genepop, file=jsm.ms.genepop.file, sep="\n")
cat(jsmackerel$popname, file=jsm.popname.file, sep=" ")

# Data load
# Prepare your GENEPOP file and population name file in the working directory
# Replace "jsm.ms.genepop.file" and "jsm.popname.file" by your file names.
popdata <- read.genepop(genepop=jsm.ms.genepop.file, popname=jsm.popname.file)

# Hedrick's G'st estimation
result.GstH <- GstH(popdata)
print(as.dist(result.GstH))
```

GstN   *Nei's Gst.*

Description

This function estimates pairwise Gst among subpopulations (Nei 1973) from a GENEPOP data object (Rousset 2008). Missing genotype values in the GENEPOP file ("0000" or "000000") are simply ignored.

Usage

```
GstN(popdata)
```

Arguments

popdata   Population data object created by read.genepop function from a GENEPOP file.

Value

Matrix of estimated pairwise Gst.

Author(s)

Reiichiro Nakamichi, Hirohisa Kishino, Shuichi Kitada

References

Nei M (1973) Analysis of Gene Diversity in Subdivided Populations. *Proc. Nat. Acad. Sci.*, 70, 3321-3323.

Rousset F (2008) Genepop'007: a complete reimplementation of the Genepop software for Windows and Linux. *Mol. Ecol. Resources*, 8, 103-106.

See Also

read.genepop

Examples

```
# Example of GENEPOP file
data(jsmackerel)
jsm.ms.genepop.file <- tempfile()
jsm.popname.file <- tempfile()
cat(jsmackerel$MS.genepop, file=jsm.ms.genepop.file, sep="\n")
cat(jsmackerel$popname, file=jsm.popname.file, sep=" ")

# Data load
# Prepare your GENEPOP file and population name file in the working directory
# Replace "jsm.ms.genepop.file" and "jsm.popname.file" by your file names.
popdata <- read.genepop(genepop=jsm.ms.genepop.file, popname=jsm.popname.file)

# Gst estimation
result.gstN <- GstN(popdata)
print(as.dist(result.gstN))
```

GstNC   *Nei and Chesser's Gst*

Description

This function estimates pairwise Gst among subpopulations (Nei&Chesser 1983) from a GENEPOP data object (Rousset 2008). Missing genotype values in the GENEPOP file ("0000" or "000000") are simply ignored.

Usage

```
GstNC(popdata)
```

Arguments

popdata   Population data object created by read.genepop function from a GENEPOP file.

Value

Matrix of estimated pairwise Gst.

Author(s)

Reiichiro Nakamichi, Hirohisa Kishino, Shuichi Kitada

References

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

Rousset F (2008) Genepop'007: a complete reimplementation of the Genepop software for Windows and Linux. *Mol. Ecol. Resources*, 8, 103-106.

See Also

read.genepop

Examples

```
# Example of GENEPOP file
data(jsmackerel)
jsm.ms.genepop.file <- tempfile()
jsm.popname.file <- tempfile()
cat(jsmackerel$MS.genepop, file=jsm.ms.genepop.file, sep="\n")
cat(jsmackerel$popname, file=jsm.popname.file, sep=" ")

# Data load
# Prepare your GENEPOP file and population name file in the working directory
# Replace "jsm.ms.genepop.file" and "jsm.popname.file" by your file names.
```

```
popdata <- read.genepop(genepop=jsm.ms.genepop.file, popname=jsm.popname.file)

# Gst estimation
result.gstNC <- GstNC(popdata)
print(as.dist(result.gstNC))
```

---

herring   *An example dataset of Atlantic herring.*

Description

An example of a genetic data for Atlantic herring population (Limborg et al. 2012). It contains genotypic information of 281 SNPs from 18 subpopulations of 607 individuals. GENEPOP format (Rousset 2008) text file is available. Subpopulation names, environmental factors (temperature and salinity) at each subpopulation and geographic distance (shortest ocean path) among subpopulations also are attached.

Usage

```
data("herring")
```

Format

$ genepop : Genotypic information of 281 SNPs in GENEPOP format text data.
$ popname : Names of subpopulations.
$ environment : Table of temperature and salinity at each subpopulation.
$ distance : Matrix of geographic distance (shortest ocean path) among subpopulations.
$ popdata : Genotype data object of this herring data created by read.genepop function.
$ fst.bootstrap : Bootstrapped Fst estimations of this herring data generated by FstBoot function.
$ fst.env : Regression analysis of environmental effects on genetic differentiation of this herring data generated by FstEnv function.

References

Limborg MT, Helyar SJ, de Bruyn M et al. (2012) Environmental selection on transcriptome-derived SNPs in a high gene flow marine fish, the Atlantic herring (Clupea harengus). *Molecular Ecology*, 21, 3686-3703.

Rousset F (2008) Genepop'007: a complete reimplementation of the Genepop software for Windows and Linux. *Mol. Ecol. Resources*, 8, 103-106.

See Also

read.genepop, FstBoot, FstEnv

Examples

```
data(herring)
ah.genepop.file <- tempfile()
ah.popname.file <- tempfile()
cat(herring$genepop, file=ah.genepop.file, sep="\n")
cat(herring$popname, file=ah.popname.file, sep=" ")

# See two text files in temporary directory.
#  ah.genepop.file  : GENEPOP format file of 281SNPs in 18 subpopulations
#  ah.popname.file  : plain text file of subpopulation names

print(herring$environment)

# herring$popdata = read.genepop(genepop="AH_genepop.txt", popname="AH_popname.txt")
# herring$fst.bootstrap = FstBoot(herring$popdata)
# herring$fst.env = FstEnv(herring$fst.bootstrap, herring$environment, herring$distance)
```

---

jsmackerel   *An example dataset of Japanese Spanich mackerel in GENEPOP and*
                                     frequency format.

Description

An example of a genetic data for a Japanese Spanish mackerel population (Nakajima et al. 2014).
It contains genotypic information of 5 microsatellite markers and mtDNA D-loop region from 8
subpopulations of 715 individuals. GENEPOP format (Rousset 2008) and frequency format (Kitada
et al. 2007) text files are available. Name list of subpopulations also is attached.

Usage

```
data("jsmackerel")
```

Format

$ MS.genepop: Genotypic information of 5 microsatellites in GENEPOP format text data.
$ MS.freq: Allele frequency of 5 microsatellites in frequency format text data.
$ mtDNA.freq: Haplotype frequency of mtDNA D-loop region in frequency format text data.
$ popname: Names of subpopulations.

Details

Frequency format file is a plain text file containing allele (haplotype) count data. This format is
mainly for mitochondrial DNA (mtDNA) haplotype frequency data, however nuclear DNA (nDNA)
data also is applicable. In the data object created by read.frequency function, "number of samples"
means haplotype count. Therefore, it equals the number of individuals in mtDNA data, however it
is the twice of the number of individuals in nDNA data. First part of the frequency format file is
the number of subpopulations, second part is the number of loci, and latter parts are [*population* x
allele] matrices of the observed allele (haplotype) counts at each locus. Two examples of frequency
format files are attached in this package.

References

Nakajima K et al. (2014) Genetic effects of marine stock enhancement: a case study based on the highly piscivorous Japanese Spanish mackerel. *Canadian Journal of Fisheries and Aquatic Sciences*, 71, 301-314.

Kitada S, Kitakado T, Kishino H (2007) Empirical Bayes inference of pairwise FST and its distribution in the genome. *Genetics*, 177, 861-873.

Rousset F (2008) Genepop'007: a complete reimplementation of the Genepop software for Windows and Linux. *Mol. Ecol. Resources*, 8, 103-106.

See Also

read.genepop, read.frequency

Examples

```
data(jsmackerel)

jsm.ms.genepop.file <- tempfile()
jsm.ms.freq.file <- tempfile()
jsm.mt.freq.file <- tempfile()
jsm.popname.file <- tempfile()
cat(jsmackerel$MS.genepop, file=jsm.ms.genepop.file, sep="\n")
cat(jsmackerel$MS.freq, file=jsm.ms.freq.file, sep="\n")
cat(jsmackerel$mtDNA.freq, file=jsm.mt.freq.file, sep="\n")
cat(jsmackerel$popname, file=jsm.popname.file, sep=" ")

# See four text files in your working directory.
#  jsm.ms.genepop.file  : GENEPOP format file of microsatellite data
#  jsm.ms.freq.file     : frequency format file of microsatellite data
#  jsm.mt.freq.file  : frequency format file of mtDNA D-loop region data
#  jsm.popname.file     : plain text file of subpopulation names
```

---

read.frequency	*Create an allele (haplotype) frequency data object of populations from a frequency format file.*

Description

This function reads a frequency format file (Kitada et al. 2007) and parse it into an R data object. This data object provides a summary of allele (haplotype) frequency in each population and marker status. This data object is used by EBFST function of this package.

Usage

```
read.frequency(frequency, popname = NULL)
```

Arguments

frequency	A character value specifying the name of the frequency format file to be analyzed.
popname	A character value specifying the name of the plain text file containing the names of subpopulations to be analyzed. This text file must not contain other than subpopulation names. The names must be separated by spaces, tabs or line breaks. If this argument is omitted, serial numbers will be assigned as subpopulation names.

Details

Frequency format file is a plain text file containing allele (haplotype) count data. This format is mainly for a mitochondrial DNA (mtDNA) haplotype frequency data, however nuclear DNA (nDNA) data also is applicable. In the data object created by read.frequency function, "number of samples" means haplotype count. Therefore, it equals the number of individuals in mtDNA data, however it is the twice of the number of individuals in nDNA data. First part of the frequency format file is the number of subpopulations, second part is the number of loci, and latter parts are [*population* x *allele*] matrices of the observed allele (haplotype) counts at each locus. Two examples of frequency format files are attached in this package. See jsmackerel.

Value

npops	Number of subpopulations.
pop_sizes	Number of samples in each subpopulation.
pop_names	Names of subpopulations.
nloci	Number of loci.
loci_names	Names of loci.
all_alleles	A list of alleles (haplotypes) at each locus.
nalleles	Number of alleles (haplotypes) at each locus.
indtyp	Number of genotyped samples in each subpopulation at each locus.
obs_allele_num	Observed allele (haplotype) counts at each locus in each subpopulation.
allele_freq	Observed allele (haplotype) frequencies at each locus in each subpopulation.
call_rate	Rate of genotyped samples at each locus.

Author(s)

Reiichiro Nakamichi, Hirohisa Kishino, Shuichi Kitada

References

Kitada S, Kitakado T, Kishino H (2007) Empirical Bayes inference of pairwise FST and its distribution in the genome. *Genetics*, 177, 861-873.

See Also

jsmackerel, EBFST

Examples

```
# Example of frequency format file
data(jsmackerel)
jsm.mt.freq.file <- tempfile()
jsm.popname.file <- tempfile()
cat(jsmackerel$mtDNA.freq, file=jsm.mt.freq.file, sep="\n")
cat(jsmackerel$popname, file=jsm.popname.file, sep=" ")

# Read frequency format file with subpopulation names
# Prepare your frequency format file and population name file in the working directory
# Replace "jsm.mt.freq.file" and "jsm.popname.file" by your file names.
popdata.mt <- read.frequency(frequency=jsm.mt.freq.file, popname=jsm.popname.file)

# Read frequency file without subpopulation names
popdata.mt.noname <- read.frequency(frequency=jsm.mt.freq.file)
```

---

read.genepop   *Create a genotype data object of populations from a GENEPOP format file.*

Description

This function reads a GENEPOP format file (Rousset 2008) and parse it into an R data object. This data object provides a summary of genotype/haplotype of each sample, allele frequency in each population, and marker status. This data object is used in downstream analysis of this package. This function is a "lite" and faster version of readGenepop function in diveRsity package (Keenan 2015).

Usage

```
read.genepop(genepop, popname = NULL)
```

Arguments

genepop   A character value specifying the name of the GENEPOP file to be analyzed.

popname   A character value specifying the name of the plain text file containing the names of subpopulations to be analyzed. This text file must not contain other than subpopulation names. The names must be separated by spaces, tabs or line breaks. If this argument is omitted, serial numbers will be assigned as subpopulation names.

Value

npops   Number of subpopulations.

pop_sizes   Number of samples in each subpopulation.

pop_names   Names of subpopulations.

nloci   Number of loci.

loci_names	Names of loci.
all_alleles	A list of alleles at each locus.
nalleles	Number of alleles at each locus.
indtyp	Number of genotyped samples in each subpopulation at each locus.
ind_names	Names of samples in each subpopulation.
pop_alleles	Genotypes of each sample at each locus in haploid designation.
pop_list	Genotypes of each sample at each locus in diploid designation.
obs_allele_num	Observed allele counts at each locus in each subpopulation.
allele_freq	Observed allele frequencies at each locus in each subpopulation.
call_rate	Rate of genotyped samples at each locus.

Author(s)

Reiichiro Nakamichi

References

Keenan K (2015) diveRsity: A Comprehensive, General Purpose Population Genetics Analysis Package. https://github.com/kkeenan02/diveRsity

Rousset F (2008) Genepop'007: a complete reimplementation of the Genepop software for Windows and Linux. *Mol. Ecol. Resources*, 8, 103-106.

Examples

```
# Example of GENEPOP file
data(jsmackerel)
jsm.ms.genepop.file <- tempfile()
jsm.popname.file <- tempfile()
cat(jsmackerel$MS.genepop, file=jsm.ms.genepop.file, sep="\n")
cat(jsmackerel$popname, file=jsm.popname.file, sep=" ")

# Read GENEPOP file with subpopulation names
# Prepare your GENEPOP file and population name file in the working directory
# Replace "jsm.ms.genepop.file" and "jsm.popname.file" by your file names.
popdata <- read.genepop(genepop=jsm.ms.genepop.file, popname=jsm.popname.file)

# Read GENEPOP file without subpopulation names
popdata.noname <- read.genepop(genepop=jsm.ms.genepop.file)
```

thetaWC.pair *Weir and Cockerham's theta adapted for pairwise Fst.*

Description

This function estimates Fst between population pairs based on Weir and Cockerham's theta (Weir & Cockerham 1984) adapted for pairwise comparison from a GENEPOP data object (Rousset 2008). Missing genotype values in the GENEPOP file ("0000" or "000000") are simply ignored.

Usage

```
thetaWC.pair(popdata)
```

Arguments

popdata   Population data object created by read.genepop function from a GENEPOP file.

Details

Weir and Cockerham (1984) derived an unbiased estimator of a coancestry coefficient (theta) based on a random effect model. It expresses the extent of genetic heterogeneity within the population. The second stage common approach is to investigate the detailed pattern of the population structure, based on a measure of genetic difference between pairs of subpopulations (demes). We call this by pairwise Fst. This function follows the formula of Weir and Cockerham's theta with the sample size $r = 2$. Given the pair, our finite sample correction multiplies a of Weir & Cockerham's theta by $(r - 1) / r$ (equation 2 in p.1359 of Weir & Cockerham 1984).

Value

Matrix of estimated pairwise Fst by theta with finite sample correction.

Author(s)

Reiichiro Nakamichi, Hirohisa Kishino, Shuichi Kitada

References

Rousset F (2008) Genepop'007: a complete reimplementation of the Genepop software for Windows and Linux. *Mol. Ecol. Resources*, 8, 103-106.

Weir BS, Cockerham CC (1984) Estimating F-statistics for the analysis of population structure. *Evolution*, 38, 1358-1370.

See Also

read.genepop

Examples

```
# Example of GENEPOP file
data(jsmackerel)
jsm.ms.genepop.file <- tempfile()
jsm.popname.file <- tempfile()
cat(jsmackerel$MS.genepop, file=jsm.ms.genepop.file, sep="\n")
cat(jsmackerel$popname, file=jsm.popname.file, sep=" ")

# Data load
# Prepare your GENEPOP file and population name file in the working directory
# Replace "jsm.ms.genepop.file" and "jsm.popname.file" by your file names.
popdata <- read.genepop(genepop=jsm.ms.genepop.file, popname=jsm.popname.file)

# theta estimation
result.theta.pair <- thetaWC.pair(popdata)
print(as.dist(result.theta.pair))
```

Index